

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science
Artificial Intelligence and Learning Systems Laboratory

Training-Free Geometric Image Editing with Diffusion Models

Diploma Thesis

by

Georgios Tzamouranis

Supervisor: Athanasios Voulodimos
Assistant Professor, N.T.U.A.

Athens, October 2026



National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science
Artificial Intelligence and Learning Systems Laboratory

Training-Free Geometric Image Editing with Diffusion Models

Diploma Thesis

by

Georgios Tzamouranis

Supervisor: Athanasios Voulodimos
Assistant Professor, N.T.U.A.

Approved by the three-member examination committee on 21 October 2026.

Athanasios Voulodimos	Georgios Stamou	Chrysoula Zerva
Assistant Professor, N.T.U.A.	Professor, N.T.U.A.	Assistant Professor, N.T.U.A.

Athens, October 2026

.....
Georgios Tzamouranis
Graduate of the School of Electrical and Computer Engineering,
National Technical University of Athens

Copyright © Georgios Tzamouranis, 2026.
All rights reserved.

You may not copy, reproduce, distribute, publish, display, modify, create derivative works, transmit, or in any way exploit this thesis or part of it for commercial purposes. You may reproduce, store or distribute this thesis for non-profit educational or research purposes, provided that the source is cited and the present copyright notice is retained. Inquiries for commercial use should be addressed to the original author.

The ideas and conclusions presented in this paper are the author's and do not necessarily reflect the official views of the National Technical University of Athens.

Περίληψη

Η γεωμετρική επεξεργασία εικόνων, δηλαδή η μετατόπιση, η περιστροφή ή η αλλαγή κλίμακας ενός αντικειμένου σε μια φωτογραφία, με παράλληλη διατήρηση της εμφάνισής του και της γύρω σκηνής, αποτελεί ιδιαίτερα απαιτητικό πρόβλημα για diffusion-based μεθόδους image editing. Το αντικείμενο πρέπει να μετασχηματιστεί χωρίς να χάσει την ταυτότητά του. Το FreeFine είναι μια training-free μέθοδος που λειτουργεί πάνω σε frozen Stable Diffusion: τοποθετεί το μετασχηματισμένο αντικείμενο σε ένα inpainted background, σχηματίζει ένα coarse composite και στη συνέχεια το περνά από στάδιο refinement. Στο GeoBench υπερτερεί έναντι προηγούμενων μεθόδων γεωμετρικής επεξεργασίας. Η παρούσα διπλωματική αναπαράγει το FreeFine, εντοπίζει συστηματικά πού και γιατί αποτυγχάνει και προτείνει βελτιώσεις χωρίς επιπλέον εκπαίδευση.

Η αναπαραγωγή εκτελεί τον δημόσιο κώδικα σε όλες τις 5.677 δισδιάστατες επεξεργασίες του GeoBench-2D. Η παραγωγή είναι ντετερμινιστική και οι μετρικές ρεαλισμού και συνέπειας συμφωνούν με τις δημοσιευμένες τιμές με απόκλιση έως 2%. Μια metric-driven διάγνωση εντοπίζει το κύριο πρόβλημα στις δύσκολες περιστροφές και στις δύσκολες αλλαγές κλίμακας, οι οποίες αποτελούν περίπου το ένα τέταρτο του benchmark και συγκεντρώνουν το μεγαλύτερο μέρος της απώλειας σε συνέπεια αντικειμένου και γεωμετρική ακρίβεια. Στη συνέχεια εξετάζονται έξι διαφορετικές εξηγήσεις της αποτυχίας μέσα από είκοσι ένα ελεγχόμενα πειράματα με κοινό πρωτόκολλο. Καμία από τις ρυθμίσεις που δοκιμάστηκαν και κανένας από τους μηχανισμούς παροχής ταυτότητας ή γεωμετρίας που εξετάστηκαν δεν εξηγεί την αποτυχία· τα ευρήματα δείχνουν, αντίθετα, προς τον τρόπο με τον οποίο το refinement χειρίζεται το coarse composite. Ο κατάλληλος χειρισμός εξαρτάται από τον μετασχηματισμό: στις μετατοπίσεις και στα non-severe resizes είναι προτιμότερο να διατηρείται το αξιόπιστο εσωτερικό του composite, στις περιστροφές το αρχικό refinement του FreeFine παραμένει η καλύτερα υποστηριζόμενη επιλογή, ενώ τα severe resizes χρειάζονται καθοδηγούμενη ανακατασκευή. Για την τελευταία περίπτωση αναπτύσσεται το EPSREC, ένας μηχανισμός guidance rectification που βασίζεται σε projected guidance και temporal memory με αρνητικό momentum και μειώνει το keypoint-position error κατά 14,7% στις δύσκολες αλλαγές κλίμακας του development subset και κατά 0,34 pixels (1,8%) στο σύνολο των 759 severe resizes του benchmark.

Τα ευρήματα συνδυάζονται στο SGR-EPSREC (Selective Geometric Refinement), μια training-free, transform-aware πολιτική που επιλέγει διαφορετικό inference-time χειρισμό με βάση μόνο τον ζητούμενο μετασχηματισμό. Στο πλήρες σύνολο των 5.677 επεξεργασιών, το SGR-EPSREC βελτιώνει αριθμητικά και τις επτά μετρικές του GeoBench-2D σε σχέση με το αναπαραγμένο FreeFine, αυξάνει τη συνέπεια αντικειμένου στο 54% των επεξεργασιών ενώ τη μειώνει στο 18%, αφήνει όλες τις περιστροφές ακριβώς όπως τις παράγει το FreeFine και δεν υποβαθμίζει τον ρεαλισμό. Η εργασία εξετάζει επίσης τι μετρούν στην πράξη οι μετρικές του benchmark (για παράδειγμα, το FreeFine φτάνει ήδη περίπου στο 99% της συνέπειας αντικειμένου του coarse composite) και διαχωρίζει τα gains που ανταναικλούν ουσιαστικά καλύτερη επεξεργασία από εκείνα που προκύπτουν εν μέρει από τον ίδιο τον ορισμό των μετρικών. Ο κώδικας, τα εκτελεσμένα notebooks και τα αποτελέσματα της εργασίας είναι δημόσια διαθέσιμα.

Λέξεις-κλειδιά: μοντέλα διάχυσης, Stable Diffusion, γεωμετρική επεξεργασία εικόνων, training-free image editing, image compositing, attention manipulation, classifier-free guidance, adaptive projected guidance, FreeFine, GeoBench, μετρικές αξιολόγησης, αναπαραγωγιμότητα

Abstract

Geometric image editing means moving, rotating or resizing an object in a photograph while keeping its appearance and the surrounding scene. It is a demanding test for diffusion-based editors, because the object must change its pose without changing its identity. FreeFine is a training-free method built on a frozen Stable Diffusion model. It places the transformed object on an inpainted background and then refines the resulting coarse composite. On the GeoBench benchmark it outperforms earlier editing methods. This thesis reproduces FreeFine, diagnoses where and why it fails, and improves it without training.

The reproduction runs the public code on all 5,677 two-dimensional edits of GeoBench-2D. It is deterministic, and it matches the published realism and consistency metrics to within 2%. A metric-driven diagnosis locates the failure in hard rotations and hard resizings, which make up a quarter of the benchmark and carry most of the loss in subject consistency and geometric precision. Six explanations of the failure are then tested in twenty-one controlled experiments under a common protocol. None of the tested reconfigurations explains the failure, and neither does any of the tested mechanisms that supply the object's identity or the exact geometry of the edit through the entry points of the frozen network. The evidence points instead to how refinement treats the coarse composite, and the right treatment depends on the transformation. For translations and moderate resizings the interior of the composite should be kept; for rotations the released refinement works best; and for severe resizings the reconstruction has to be guided. For this last case the thesis develops EPSREC, a projected guidance with recursive momentum. It lowers the keypoint error by 14.7% on the hard resizings of the development subset, and by 0.34 pixels (1.8%) on all 759 severe resizings of the benchmark.

These findings are combined into SGR-EPSREC (Selective Geometric Refinement), a training-free policy that reads only the requested transformation and sends each edit to the treatment that suits it. On all 5,677 edits, SGR-EPSREC improves all seven metrics of GeoBench-2D over FreeFine. It raises subject consistency for 54% of the edits and lowers it for 18%, leaves every rotation exactly as FreeFine produces it, and does not lower realism. The thesis also examines what the benchmark's metrics can and cannot measure. FreeFine already reaches 99% of the subject consistency of the composite, which is a copy of the source at its target pose, and part of the measured gains follows from how the metrics are built. On this basis, the thesis separates the improvements that reflect better editing from those that do not. All code, executed notebooks and results are publicly available.

Keywords: diffusion models, Stable Diffusion, geometric image editing, training-free editing, image composition, attention manipulation, classifier-free guidance, adaptive projected guidance, FreeFine, GeoBench, evaluation metrics, reproducibility

Acknowledgements

I would first like to thank my supervisor, Asst. Prof. Athanasios Voulodimos, and Prof. Georgios Stamou, for giving me the opportunity to work on this subject and for their guidance, support and trust throughout my studies. I am also grateful to the Artificial Intelligence and Learning Systems Laboratory for providing the environment in which this thesis was carried out and for the inspiration it offered.

I owe special thanks to Ilias Mitsouras and Vasileios Karampinis, the PhD candidates who advised me throughout this work.

Finally, I owe my deepest gratitude to my family. Their love, patience, unconditional support and unwavering belief in me throughout all the years of my studies have been the foundation of everything I have achieved. I also thank my friends for their encouragement, for the countless days of studying we shared, for making the difficult moments feel lighter by facing them together, and for the laughter that made even the most demanding times memorable.

Contents

Περίληψη **5**

Abstract **7**

Acknowledgements **9**

List of Figures **17**

List of Tables **19**

List of Abbreviations **21**

1 Εκτεταμένη Περίληψη στα Ελληνικά **23**
1.1 Εισαγωγή και στόχοι . . . 23
1.2 Θεωρητικό υπόβαθρο . . . 24
1.2.1 Μοντέλα διάχυσης και Stable Diffusion . . . 24
1.2.2 Η μέθοδος FreeFine . . . 25
1.2.3 Το GeoBench-2D και οι μετρικές αξιολόγησης . . . 25
1.3 Αναπαραγωγή και διάγνωση . . . 26
1.4 Οι υποθέσεις και τα πειράματα . . . 27
1.4.1 Υ1: ρύθμιση της μεθόδου (Πειράματα 3–7) . . . 27
1.4.2 Υ2 και Υ3: πληροφορία που λείπει από το μοντέλο (Πειράματα 8–11) . . . 28
1.4.3 Υ4: αγκύρωση στο coarse composite (Πειράματα 12–17) . . . 29
1.4.4 Υ5: guidance για severe resizes (Πειράματα 18–21) . . . 30
1.4.5 Υ6: δομή του composite μέσα στο αντικείμενο (Πειράματα 22–23) . . . 31
1.4.6 Το κοινό μοτίβο . . . 31
1.5 Η προτεινόμενη μέθοδος . . . 32
1.5.1 Αποκατάσταση του εσωτερικού . . . 32
1.5.2 EPSREC . . . 32
1.5.3 Ο κανόνας routing και το SGR-EPSREC . . . 32
1.6 Αποτελέσματα στο πλήρες σύνολο . . . 33
1.7 Συμπεράσματα . . . 34

2 Introduction **35**
2.1 Geometric image editing . . . 35
2.2 Training-free editing on frozen diffusion models . . . 35
2.3 Problem statement and research questions . . . 36
2.4 Contributions . . . 36
2.5 Scope and non-goals . . . 38
2.6 Structure of the thesis . . . 38

3 Diffusion Models and Stable Diffusion **41**
3.1 Denoising diffusion probabilistic models . . . 41

3.2 Deterministic sampling with DDIM . . . 43
3.3 DDIM inversion . . . 44
3.4 Partial denoising and editing strength . . . 44
3.5 Latent diffusion and Stable Diffusion 1.5 . . . 45
3.6 Attention in the U-Net . . . 47
3.7 Classifier-free guidance . . . 48

4 Related Work **51**
4.1 Training-free editing with diffusion models . . . 51
4.2 Drag-based and point-based editing . . . 52
4.3 Geometric and object-level editing . . . 52
4.4 Conditioning a frozen model . . . 53
4.5 Guidance beyond classifier-free guidance . . . 54
4.6 Feature-space and frequency-domain manipulation . . . 54
4.7 Evaluating geometric editing . . . 54
4.8 Positioning of this thesis . . . 55

5 FreeFine and GeoBench-2D **57**
5.1 Task and notation . . . 57
5.2 The decoupled pipeline . . . 57
5.3 Object transformation and the coarse composite . . . 58
5.4 Refinement . . . 59
5.4.1 Inversion and the four branches . . . 59
5.4.2 Temporal Contextual Attention . . . 60
5.4.3 Local Perturbation . . . 60
5.4.4 Content-specified Generation . . . 61
5.5 The released configuration . . . 61
5.6 GeoBench-2D . . . 62
5.7 The evaluation metrics . . . 63
5.8 Properties of the metric implementation . . . 63

6 Methodology **65**
6.1 Problem statement and research questions . . . 65
6.2 Overview of the study . . . 66
6.3 The six hypotheses and how each is tested . . . 68
6.4 The method in overview . . . 70
6.5 Evaluation design . . . 72
6.6 Implementation and reproducibility . . . 73

7 Reproduction and Diagnosis **75**
7.1 Experiment 1: Reproducing FreeFine . . . 75
7.1.1 What was done . . . 75
7.1.2 Results . . . 77
7.1.3 Outcome . . . 78
7.2 Experiment 2: Where FreeFine fails . . . 79
7.2.1 What was done . . . 79
7.2.2 Results . . . 80
7.2.3 Outcome . . . 80

8 Experimental Protocol **87**
8.1 Development and held-out subsets . . . 87

8.2 Groups . . . 88
8.3 Controls that accompany every study . . . 88
8.4 The stochastic metrics . . . 89
8.5 Reading the results . . . 89
8.6 Provenance . . . 90

9 Hypothesis 1: Tuning the Released Pipeline 91
9.1 Experiment 3: Refinement strength . . . 91
9.1.1 Change . . . 91
9.1.2 Results . . . 92
9.1.3 Outcome . . . 92
9.2 Experiment 4: The layer range of TCA . . . 93
9.2.1 Change . . . 93
9.2.2 Results . . . 94
9.2.3 Outcome . . . 94
9.3 Experiment 5: Guidance with the object label . . . 95
9.3.1 Change . . . 95
9.3.2 Results . . . 95
9.3.3 Outcome . . . 96
9.4 Experiment 6: The floor of the TCA schedule . . . 96
9.4.1 Change . . . 96
9.4.2 Results . . . 97
9.4.3 Outcome . . . 98
9.5 Experiment 7: An anti-aliased coarse composite . . . 98
9.5.1 Change . . . 98
9.5.2 Results . . . 99
9.5.3 Outcome . . . 99
9.6 The configuration space as a whole . . . 100

10 Hypotheses 2 and 3: Supplying Information the Model Lacks 103
10.1 Experiment 8: Identity through image tokens . . . 103
10.1.1 Change . . . 103
10.1.2 Results . . . 105
10.1.3 Outcome . . . 105
10.2 Experiment 9: Geometry through image tokens . . . 106
10.2.1 Change . . . 106
10.2.2 Results . . . 107
10.2.3 Outcome . . . 107
10.3 Experiment 10: Geometry through residual feature maps . . . 108
10.3.1 Change . . . 108
10.3.2 Results . . . 108
10.3.3 Outcome . . . 109
10.4 Experiment 11: Geometry inside the attention transplant . . . 109
10.4.1 Change . . . 109
10.4.2 Results . . . 114
10.4.3 Outcome . . . 115
10.5 The entry points as a whole . . . 117

11 Hypothesis 4: Anchoring the Refinement to the Coarse Composite 119
11.1 Experiment 12: Anchoring during denoising . . . 119
11.1.1 Change . . . 119

11.1.2 Results . . . 120
11.1.3 Outcome . . . 120
11.2 Experiment 13: The composite as an output . . . 121
11.2.1 Change . . . 121
11.2.2 Results . . . 121
11.2.3 Outcome . . . 122
11.3 Experiment 14: Routed models with interior restoration . . . 123
11.3.1 Change . . . 123
11.3.2 Results . . . 124
11.3.3 Outcome . . . 125
11.4 Experiment 15: The width of the refined band . . . 125
11.4.1 Change . . . 125
11.4.2 Results . . . 126
11.4.3 Outcome . . . 126
11.5 Experiment 16: What makes interior restoration work . . . 127
11.5.1 Change . . . 127
11.5.2 Results . . . 127
11.5.3 Outcome . . . 128
11.6 Experiment 17: Interior restoration for each edit type . . . 128
11.6.1 Change . . . 128
11.6.2 Results . . . 129
11.6.3 Outcome . . . 129

12 Hypothesis 5: Guiding the Reconstruction of Severe Resizings 131
12.1 Experiment 18: Projected guidance in ε-space . . . 132
12.1.1 Change . . . 132
12.1.2 Results . . . 133
12.1.3 Outcome . . . 133
12.2 Experiment 19: Frequency-decoupled guidance . . . 134
12.2.1 Change . . . 134
12.2.2 Results . . . 136
12.2.3 Outcome . . . 136
12.3 Experiment 20: APG in D-space and the EPSREC formulation . . . 137
12.3.1 Change . . . 137
12.3.2 Results . . . 142
12.3.3 Outcome . . . 143
12.4 Experiment 21: Where the guidance should act . . . 144
12.4.1 Change . . . 144
12.4.2 Results . . . 145
12.4.3 Outcome . . . 145

13 Hypothesis 6: Supplying the Composite's Structure inside the Object 147
13.1 Experiment 22: Spectral feature fusion . . . 148
13.1.1 Change . . . 148
13.1.2 Results . . . 150
13.1.3 Outcome . . . 151
13.2 Experiment 23: The spectral search . . . 151
13.2.1 Change . . . 151
13.2.2 Results . . . 152
13.2.3 Outcome . . . 153

14 Selective Geometric Refinement **155**
14.1 The four branches . . . 155
14.2 Experiment 24: A routing rule from the requested transformation . . . 156
14.2.1 Change . . . 156
14.2.2 Results . . . 157
14.2.3 Outcome . . . 157
14.3 SGR-EPSREC . . . 158
14.4 Experiment 25: The router on the development subset . . . 161
14.4.1 Change . . . 161
14.4.2 Results . . . 161
14.4.3 Outcome . . . 162

15 Full-Scale Evaluation **165**
15.1 Evaluation setup . . . 165
15.2 Results . . . 165
15.3 How much of the improvement is real . . . 169
15.4 A gallery of edits . . . 170
15.5 Summary . . . 170

16 Discussion and Limitations **175**
16.1 What the six hypotheses establish . . . 175
16.2 How much room there was . . . 177
16.3 What the metrics can show . . . 177
16.4 Limitations . . . 178
16.5 Generality . . . 179

17 Conclusions and Future Work **181**
17.1 Conclusions . . . 181
17.2 Future work . . . 182

Bibliography **185**

List of Figures

2.1 The structure of the thesis 38

3.1 The diffusion process 42
3.2 The noise schedule of Stable Diffusion 1.5 44
3.3 Stable Diffusion 1.5 46

5.1 The pipeline on one edit 58
5.2 The decoupled pipeline 59
5.3 The TCA blending schedule 61

6.1 The stages of the study 66
6.2 Where each hypothesis intervenes 68

7.1 The artefacts of the reproduction 77
7.2 Reproduction against the published values 78
7.3 The baseline by edit type and severity 81
7.4 Keypoint error by edit type and severity 82
7.5 Realism metrics across groups 83
7.6 FreeFine on an easy translation and a hard rotation 84
7.7 FreeFine on a severe reduction and a severe enlargement 85

9.1 The start-step sweep 94
9.2 TCA schedules with a raised floor 97
9.3 Every configuration change against the baseline 100

10.1 Where information can enter a frozen U-Net 104
10.2 How TCA's masks reach the attention heads 110
10.3 Where the geometry enters TCA 111
10.4 The weight profile of the correspondence bias 112
10.5 Geometry inside attention against the per-branch layout 117

11.1 The composite against the refined output 122
11.2 Interior restoration 123
11.3 The width of the refined band 126
11.4 Interior restoration on two edits 130

12.1 Where the reconstruction is steered 132
12.2 Frequency-decoupled guidance 135
12.3 The factor c_t and the weights of past corrections 138
12.4 Guidance formulations on hard resizings 143

13.1 Spectral fusion with and without label guidance 154

14.1 The SGR-EPSREC router 158

14.2 FreeFine and SGR-EPSREC: translation and moderate resizing 159
14.3 FreeFine and SGR-EPSREC: rotation and severe resizing 160

15.1 Typical edits of the full benchmark 167
15.2 The largest losses in subject consistency 168
15.3 Per-edit change in subject consistency 169
15.4 Gallery of edits: translations (1) 171
15.5 Gallery of edits: translations (2) 172
15.6 Gallery of edits: moderate resizings 173

16.1 What the method leaves unsolved 180

List of Tables

1.1 Οι επτά μετρικές αξιολόγησης του GeoBench-2D . . . 26
1.2 Αναπαραγωγή του FreeFine σε σύγκριση με τις δημοσιευμένες τιμές . . . 26
1.3 Οι έξι υποθέσεις, οι παρεμβάσεις και τα συμπεράσματα . . . 28
1.4 Οι τέσσερις branches του SGR-EPSREC . . . 33
1.5 FreeFine και SGR-EPSREC στο πλήρες GeoBench-2D . . . 33

3.1 The transformer blocks of SD-1.5 . . . 47
3.2 Notation for diffusion models . . . 49

4.1 Methods closest to this thesis . . . 55

5.1 Notation for the pipeline . . . 58
5.2 Released settings of the 2D benchmark script . . . 61
5.3 Composition of GeoBench-2D . . . 62
5.4 The seven metrics . . . 64

6.1 The stages of the study . . . 67
6.2 The six hypotheses and their tests . . . 69
6.3 The three scales of evaluation . . . 72

7.1 The two pinned environments . . . 76
7.2 Changes required to run the released code . . . 76
7.3 Reproduction against the published results . . . 77
7.4 Four evaluations of the same images . . . 78
7.5 Severity tertiles against edit type . . . 80
7.6 The baseline by edit type . . . 80
7.7 The baseline by severity tertile . . . 81

8.1 Composition of the development subset . . . 88
8.2 The standard groups . . . 88

9.1 Settings varied in this chapter . . . 92
9.2 What the start step sets . . . 92
9.3 The start-step sweep . . . 93
9.4 The layer range of TCA . . . 95
9.5 Guidance with the object label . . . 96
9.6 The floor of the TCA schedule . . . 97
9.7 The anti-aliased warp . . . 99
9.8 Summary of the configuration space . . . 101

10.1 The IP-Adapter scale sweep . . . 105
10.2 Geometry through image tokens . . . 107
10.3 Geometry through ControlNet . . . 109
10.4 Head-layout test on the released code . . . 111

10.5 The geometry-aware mechanisms 114
10.6 Verification of the implementation 115
10.7 Geometry inside attention 116
10.8 What each entry point achieved 118

11.1 Latent preservation 120
11.2 The coarse composite as an output 121
11.3 Routed models with interior restoration 124
11.4 The width of the refined band 126
11.5 Scope of the anchoring 127
11.6 Warp quality and interior restoration 127
11.7 Interior restoration for each edit type 128

12.1 Projected guidance in ε-space 133
12.2 Frequency-decoupled guidance 136
12.3 Weights of past corrections 141
12.4 Settings of the APG variants 141
12.5 Results of the APG variants 142
12.6 The scope of the guidance 145

13.1 Spectral feature fusion 150
13.2 The spectral search 152
13.3 Spectral fusion combined with EPSREC 153

14.1 The branches of the router 156
14.2 Scale ranges of resizing by difficulty 157
14.3 The routing rule on all resizings 157
14.4 The routing rule against the severity tertiles 158
14.5 The router on the development subset 162
14.6 Realism of the router on the development subset 162
14.7 The method and its ablation 163

15.1 The method on the whole benchmark 166
15.2 The method by branch 166

16.1 The six hypotheses 176

List of Abbreviations

APG	Adaptive Projected Guidance
BGC	Background Consistency
CFG	Classifier-Free Guidance
CG	Content-specified Generation
DDIM	Denoising Diffusion Implicit Model
DDPM	Denoising Diffusion Probabilistic Model
DIFT	Diffusion Features
FID	Frechet Inception Distance
FDD	Frechet DINOv2 Distance
KD	Kernel Distance
LP	Local Perturbation
MD	Mean keypoint Distance
SGR	Selective Geometric Refinement
SUBC	Subject Consistency
TCA	Temporal Contextual Attention
WRAP_E	Warping Error

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

1.1 Εισαγωγή και στόχοι

Η γεωμετρική επεξεργασία εικόνων απαιτεί από ένα παραγωγικό μοντέλο να μετακινήσει, να περιστρέψει ή να αλλάξει το μέγεθος ενός αντικειμένου μέσα σε μια φωτογραφία, έτσι ώστε το αντικείμενο να εμφανίζεται στη νέα ζητούμενη θέση, να παραμένει αναγνωρίσιμο το ίδιο και η υπόλοιπη σκηνή να διατηρείται όσο το δυνατόν αμετάβλητη.

Το πρόβλημα συνδυάζει απαιτήσεις που συχνά συγκρούονται μεταξύ τους. Το αντικείμενο πρέπει να διατηρήσει την εμφάνιση και την ταυτότητά του· η περιοχή που αφήνει πίσω του πρέπει να συμπληρωθεί με πειστικό background· το αντικείμενο πρέπει να ενσωματωθεί φυσικά στη νέα του θέση χωρίς εμφανείς ραφές ή artifacts· και, σε ορισμένες περιπτώσεις, πρέπει να παραχθεί οπτική πληροφορία που δεν υπήρχε στην αρχική εικόνα. Η απλή αντιγραφή των pixels προστατεύει την ταυτότητα του αντικειμένου, αλλά δυσκολεύεται να το εντάξει φυσικά στη νέα σκηνή. Αντίθετα, η πλήρης αναπαραγωγή του από το generative model μπορεί να δώσει πιο φυσικό αποτέλεσμα, αλλά αυξάνει τον κίνδυνο να αλλοιωθεί η ταυτότητά του.

Τα μοντέλα διάχυσης (diffusion models) [2, 8] έχουν βελτιώσει σημαντικά την ποιότητα της παραγωγής εικόνων και αποτελούν πλέον βασικό εργαλείο για image editing. Μια σημαντική κατηγορία training-free μεθόδων αξιοποιεί τη γνώση που είναι ήδη αποθηκευμένη στα frozen weights μεγάλων pretrained μοντέλων, όπως το Stable Diffusion, χωρίς να εκπαιδεύει νέες παραμέτρους. Οι μέθοδοι αυτές συνήθως εφαρμόζουν inversion της πραγματικής εικόνας στον latent/noise space και ελέγχουν την επαναπαραγωγή της μέσω attention manipulation, guidance ή τοπικού ελέγχου του denoising. Για γεωμετρικές επεξεργασίες, η ισχυρότερη από τις μεθόδους που αξιολογούνται στο GeoBench είναι το FreeFine [34]. Το FreeFine αποσυνδέει το πρόβλημα σε τρία στάδια: μετασχηματισμό του αντικειμένου, inpainting της περιοχής που αυτό αφήνει πίσω του και refinement του coarse composite που προκύπτει. Η διατήρηση της εμφάνισης του αντικειμένου επιτυγχάνεται με μεταφορά χαρακτηριστικών από την πηγή μέσω attention. Οι δημιουργοί του FreeFine εισήγαγαν επίσης το benchmark GeoBench, στο οποίο η μέθοδος υπερτερεί έναντι οκτώ προηγούμενων προσεγγίσεων.

Η δημοσιευμένη αξιολόγηση του FreeFine βασίζεται κυρίως σε μέσους όρους πάνω στο σύνολο των edits. Οι μέσοι αυτοί δεν δείχνουν σε ποιες περιπτώσεις αποτυγχάνει η μέθοδος, ποια είναι η αιτία της αποτυχίας και αν το πρόβλημα μπορεί να αντιμετωπιστεί χωρίς να εγκαταλειφθεί ο training-free χαρακτήρας της. Η παρούσα εργασία μελετά το FreeFine σε όλες τις 5.677 δισδιάστατες επεξεργασίες του GeoBench-2D και οργανώνεται γύρω από τρία ερευνητικά ερωτήματα:

1. Μπορούν τα αποτελέσματα του FreeFine να αναπαραχθούν πιστά από τον δημόσιο κώδικά του και σε ποιες κατηγορίες edits εμφανίζονται οι μεγαλύτερες αποτυχίες;

2. Πού οφείλεται η αποτυχία: σε suboptimal configuration της υπάρχουσας μεθόδου, σε πληροφορία που δεν διαθέτει το frozen model ή στον τρόπο με τον οποίο το refinement χειρίζεται το coarse composite;

3. Μπορεί το FreeFine να βελτιωθεί χωρίς επιπλέον εκπαίδευση, και σε ποιο βαθμό, όταν διαχωριστούν τα gains που αντιστοιχούν σε ουσιαστικά καλύτερη επεξεργασία από εκείνα που προκύπτουν από τον ορισμό των ίδιων των μετρικών;

Οι κύριες συνεισφορές της εργασίας είναι έξι:

1. **Πιστή και ντετερμινιστική αναπαραγωγή του FreeFine** σε όλες τις 5.677 επεξεργασίες, με απόκλιση έως 2% στις δημοσιευμένες μετρικές ρεαλισμού και συνέπειας, καθώς και ανάλυση του released code ώστε να αποσαφηνιστεί πώς λειτουργούν στην πράξη οι βασικές παράμετροι της μεθόδου.

2. **Metric-driven διάγνωση και ελεγχόμενο πειραματικό πρωτόκολλο**, που εντοπίζουν το κύριο πρόβλημα στις δύσκολες περιστροφές και αλλαγές κλίμακας και ορίζουν με σαφήνεια τους όρους κάθε μεταγενέστερης σύγκρισης.

3. **Συστηματικός έλεγχος έξι διαφορετικών εξηγήσεων της αποτυχίας**, μέσα από είκοσι ένα ελεγχόμενα πειράματα. Καμία από τις ρυθμίσεις και κανένας από τους μηχανισμούς παροχής ταυτότητας ή γεωμετρίας που δοκιμάστηκαν δεν εξηγεί την αποτυχία· τα ευρήματα δείχνουν, αντίθετα, προς τον τρόπο με τον οποίο το refinement χειρίζεται το coarse composite, και ότι ο κατάλληλος χειρισμός εξαρτάται από τον ίδιο τον μετασχηματισμό.

4. **Το EPSREC**, ένας μηχανισμός guidance rectification για severe resizes, ο οποίος μειώνει το keypoint-position error κατά 14,7% στις δύσκολες αλλαγές κλίμακας του development subset και κατά 0,34 pixels (1,8%) στο σύνολο των 759 severe resizes του benchmark.

5. **Το SGR-EPSREC**, μια training-free, transform-aware πολιτική refinement που επιλέγει διαφορετικό χειρισμό ανά edit και βελτιώνει αριθμητικά και τις επτά μετρικές στο πλήρες σύνολο των 5.677 επεξεργασιών.

6. **Ανάλυση του τι μετρούν πραγματικά οι μετρικές του benchmark**, ώστε να διαχωριστούν οι βελτιώσεις που αντανακλούν ουσιαστικά καλύτερο editing από εκείνες που προκύπτουν εν μέρει από την κατασκευή των μετρικών.

1.2 Θεωρητικό υπόβαθρο

1.2.1 Μοντέλα διάχυσης και Stable Diffusion

Ένα diffusion model μαθαίνει να αντιστρέφει μια διαδικασία κατά την οποία τα δεδομένα αλλοιώνονται σταδιακά με Gaussian noise. Στη forward process προστίθεται θόρυβος σε T βήματα και, σε κλειστή μορφή, ισχύει:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathrm{I}), \tag{1.1}$$

όπου ο συντελεστής $\bar{\alpha}_t$ μειώνεται από το 1 προς το 0. Ένα νευρωνικό δίκτυο $\varepsilon_\theta(z_t, t, c)$ εκπαιδεύεται να προβλέπει τον θόρυβο ε, υπό μια συνθήκη c, για παράδειγμα ένα text prompt. Από την πρόβλεψη του

θορύβου προκύπτει η αντίστοιχη εκτίμηση του clean sample:

$$D = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon_\theta}{\sqrt{\bar{\alpha}_t}}. \tag{1.2}$$

Ο deterministic sampler DDIM [5] μεταβαίνει από το z_t στο $z_{t'}$ συνδυάζοντας την πρόβλεψη του clean sample με την πρόβλεψη του θορύβου, $z_{t'} = \sqrt{\bar{\alpha}_{t'}}\,D + \sqrt{1-\bar{\alpha}_{t'}}\,\varepsilon_\theta$. Επειδή η διαδικασία είναι ντετερμινιστική, η ίδια τροχιά μπορεί να ακολουθηθεί προσεγγιστικά και προς την αντίθετη κατεύθυνση μέσω DDIM inversion: μια πραγματική εικόνα χαρτογραφείται σε ένα noisy latent και στη συνέχεια ανακατασκευάζεται από αυτό. Η ιδιότητα αυτή αποτελεί τη βάση πολλών training-free editing μεθόδων.

Το Stable Diffusion [8] εφαρμόζει τη diffusion process σε latent space. Ένας autoencoder μετατρέπει μια εικόνα 512×512 σε latent αναπαράσταση $64 \times 64 \times 4$, ενώ ένα U-Net με self-attention και cross-attention προς το κείμενο εκτελεί το denoising. Η classifier-free guidance (CFG) [19] ενισχύει την επίδραση του conditioning, κάνοντας extrapolation από την unconditional πρόβλεψη $\varepsilon_\varnothing$ προς την conditional πρόβλεψη ε_p:

$$\hat{\varepsilon} = \varepsilon_\varnothing + w\,(\varepsilon_p - \varepsilon_\varnothing). \tag{1.3}$$

1.2.2 Η μέθοδος FreeFine

Το FreeFine υλοποιεί μια γεωμετρική επεξεργασία σε τρία στάδια (Κεφάλαιο 5). Αρχικά εφαρμόζεται ο ζητούμενος μετασχηματισμός στο αντικείμενο, γύρω από το κέντρο του bounding box του, και προκύπτει η μάσκα στόχου M_t. Στη συνέχεια γίνεται inpainting της περιοχής που άφησε κενή το αντικείμενο. Το μετασχηματισμένο αντικείμενο τοποθετείται πάνω στο inpainted background και σχηματίζει το coarse composite I_c. Τέλος, το composite περνά από refinement: γίνεται DDIM inversion και partial denoising μόνο στα τελευταία 15 από τα 50 βήματα, ώστε να διατηρείται η βασική γεωμετρία της εισόδου ενώ διορθώνονται seams και artifacts του resampling.

Το refinement χρησιμοποιεί τρεις βασικούς μηχανισμούς πάνω στο frozen Stable Diffusion 1.5. Η Temporal Contextual Attention (TCA) επιτρέπει στα queries του edit branch να αντλούν keys και values από το reference branch, με βάρος που μειώνεται σταδιακά κατά τη διάρκεια του refinement· έτσι διατηρούνται η εμφάνιση και τα χαρακτηριστικά της πηγής. Η Local Perturbation προσθέτει στοχαστικό θόρυβο μόνο σε συγκεκριμένες περιοχές γύρω από το αντικείμενο και στην περιοχή που έχει αδειάσει, ώστε εκεί να μπορεί να παραχθεί νέο περιεχόμενο. Τέλος, η Content-specified Generation χρησιμοποιεί το όνομα του αντικειμένου ως prompt και εφαρμόζει local classifier-free guidance σε μια ζώνη γύρω από αυτό. Στη released ρύθμιση, όμως, το prompt είναι κενό, επομένως η guidance term δεν έχει πρακτική επίδραση.

1.2.3 Το GeoBench-2D και οι μετρικές αξιολόγησης

Το GeoBench-2D περιλαμβάνει 5.677 edits πάνω σε 609 εικόνες: 1.439 μετατοπίσεις, 1.603 περιστροφές έως 39° και 2.635 ισότροπες αλλαγές κλίμακας με scale factor από 0,40 έως 2,88. Για κάθε edit παρέχονται η μάσκα του αντικειμένου και οι ακριβείς παράμετροι του μετασχηματισμού. Οι επτά μετρικές του benchmark (Πίνακας 1.1) καλύπτουν τρεις άξονες: τον ρεαλισμό των παραγόμενων εικόνων ως κατανομή, τη συνέπεια του edited αποτελέσματος με την εικόνα πηγής και τη γεωμετρική ακρίβεια της εκτέλεσης του μετασχηματισμού.

Πίνακας 1.1: Οι επτά μετρικές αξιολόγησης του GeoBench-2D.

Μετρική	Τι μετρά	Κατεύθυνση
FID, FID_{dino}, KD	Απόσταση της κατανομής των edited εικόνων από την κατανομή των πραγματικών εικόνων (distributional realism).	↓
SUBC	Ομοιότητα του edited αντικειμένου με το αρχικό, με DINO features (subject consistency).	↑
BGC	Ομοιότητα του background έξω από τις source/target masks, με CLIP features (background consistency).	↑
WRAP_E	Απόσταση της εξόδου από το coarse composite μέσα στη μάσκα στόχου (warping error).	↓
MD	Μέση απόσταση των keypoints του αντικειμένου από τις θέσεις-στόχους που ορίζει ο μετασχηματισμός, σε pixels.	↓

Πίνακας 1.2: Αναπαραγωγή του FreeFine σε σύγκριση με τις δημοσιευμένες τιμές, στο πλήρες σύνολο των 5.677 edits.

Μετρική	Δημοσιευμένη τιμή	Αναπαραγωγή	Διαφορά
FID↓	34,72	35,04	+0,9%
FID_{dino}↓	478,18	487,82	+2,0%
KD↓	0,144	0,142	−1,2%
SUBC↑	0,907	0,911	+0,5%
BGC↑	0,971	0,967	−0,4%
WRAP_E↓	0,055	0,047	−13,9%
MD↓	9,25	8,50	−8,1%

1.3 Αναπαραγωγή και διάγνωση

Αναπαραγωγή (Πείραμα 1). Ο δημόσιος κώδικας του FreeFine εκτελέστηκε σε όλες τις 5.677 επεξεργασίες, με μόνο τις αλλαγές που ήταν απαραίτητες για το περιβάλλον εκτέλεσης και χωρίς καμία αλγοριθμική τροποποίηση. Η παραγωγή είναι ντετερμινιστική: επαναλαμβανόμενα runs παράγουν byte-identical εικόνες, γεγονός που επιτρέπει να γίνονται όλες οι επόμενες συγκρίσεις edit προς edit. Ο Πίνακας 1.2 συγκρίνει την αναπαραγωγή με τα δημοσιευμένα αποτελέσματα. Οι μετρικές ρεαλισμού και συνέπειας συμφωνούν με απόκλιση έως 2%, ενώ οι WRAP_E και MD προκύπτουν καλύτερες από τις δημοσιευμένες τιμές. Εφόσον η αιτία αυτής της διαφοράς δεν εντοπίστηκε με βεβαιότητα, όλες οι μεταγενέστερες συγκρίσεις της εργασίας γίνονται απέναντι στο reproduced baseline και όχι απευθείας απέναντι στους αριθμούς του paper. Η MD αποδείχθηκε στοχαστική μετρική: η επαναξιολόγηση των ίδιων εικόνων μετέβαλε την τιμή της κατά 0,046, οπότε σε όλα τα επόμενα πειράματα υπολογίζεται με σταθερό random seed.

Διάγνωση (Πείραμα 2). Τα edits χωρίστηκαν ανά τύπο μετασχηματισμού και ανά τριτημόριο δυσκολίας. Η SUBC μειώνεται από 0,942 στο εύκολο τριτημόριο σε 0,865 στο δύσκολο, ενώ η MD σχεδόν τριπλασιάζεται, από 4,86 σε 13,51 pixels. Αντίθετα, η BGC και η WRAP_E παραμένουν σχεδόν σταθερές. Η βασική αστοχία αφορά επομένως τη δομή και τη γεωμετρική συνέπεια του αντικειμένου, όχι το background, και συγκεντρώνεται στις δύσκολες περιστροφές και στις δύσκολες αλλαγές κλίμακας: 1.480 edits, δηλαδή περίπου το 26% του συνόλου, με σχεδόν διπλάσιο keypoint error από τις υπόλοιπες περιπτώσεις. Η διάγνωση ανέδειξε επίσης ιδιότητες του official metric code που είναι κρίσιμες για την ερμηνεία των αποτελεσμάτων: οι distributional realism metrics είναι συγκρίσιμες μόνο μεταξύ μεθόδων που αξιολογούνται στην ίδια ομάδα, η MD υπολογίζει τον στόχο των keypoints με διαφορετική σύμβαση

κέντρου από εκείνη που χρησιμοποιεί το transform implementation, και κάθε μέθοδος αξιολογείται πάνω στα keypoints που ανιχνεύονται στη δική της έξοδο.

Πρωτόκολλο. Κάθε υπόθεση ελέγχθηκε σε ένα stratified development subset 200 edits, ισορροπημένο ως προς τον τύπο μετασχηματισμού και τη δυσκολία. Ένα disjoint held-out subset άλλων 200 edits χρησιμοποιήθηκε ως δεύτερο development check για τα πρώτα combined models. Κάθε τροποποίηση υλοποιήθηκε πίσω από ξεχωριστό switch και, πριν από κάθε μελέτη, επαληθεύτηκε ότι με τον μηχανισμό απενεργοποιημένο ο κώδικας αναπαράγει τις αποθηκευμένες baseline εικόνες με μέση απόλυτη διαφορά 0,0. Με αυτόν τον τρόπο, κάθε μετρούμενη διαφορά μπορεί να αποδοθεί στην εξεταζόμενη αλλαγή. Ως πρακτικό threshold για meaningful βελτίωση της SUBC σε δύσκολη υποομάδα ορίστηκε το +0,005, περίπου διπλάσιο από τη μεγαλύτερη μεταβολή που παρατηρήθηκε μόνο μέσω tuning της αρχικής μεθόδου.

1.4 Οι υποθέσεις και τα πειράματα

Η αποτυχία που ανέδειξε η διάγνωση μπορεί να εξηγείται από διαφορετικούς μηχανισμούς. Για τον λόγο αυτό διατυπώθηκαν έξι υποθέσεις και καθεμία ελέγχθηκε με παρεμβάσεις που στοχεύουν άμεσα στη συγκεκριμένη αιτία. Συνολικά πραγματοποιήθηκαν είκοσι ένα ελεγχόμενα πειράματα (Πειράματα 3–23), όλα στο development subset και με το κοινό πρωτόκολλο της Ενότητας 1.3. Ο Πίνακας 1.3 συνοψίζει τις έξι υποθέσεις, τις αλλαγές που εξετάστηκαν και το συμπέρασμα κάθε μελέτης. Οι επόμενες υποενότητες παρουσιάζουν τα πειράματα με τη σειρά που πραγματοποιήθηκαν.

1.4.1 Υ1: ρύθμιση της μεθόδου (Πειράματα 3–7)

Η πρώτη υπόθεση είναι ότι το FreeFine δεν χρειάζεται νέο μηχανισμό, αλλά καλύτερο tuning των υπαρχουσών ρυθμίσεών του. Πέντε πειράματα μεταβάλλουν, μία κάθε φορά, βασικές παραμέτρους του released pipeline και την ποιότητα του coarse input.

- **Πείραμα 3: refinement strength.** Το start step s καθορίζει σε ποιο σημείο του 50-step DDIM schedule ξεκινά το denoising. Επομένως ελέγχει πόσο από το coarse composite επιτρέπεται να ξαναγραφτεί, πόσα refinement steps εκτελούνται και για πόσα βήματα παραμένει ενεργή η TCA. Δοκιμάστηκαν $s = 25$, 30, 40 και 45 γύρω από τη δημοσιευμένη τιμή $s = 35$.

- **Πείραμα 4: TCA layer range.** Η παράμετρος που υποτίθεται ότι όριζε το πρώτο layer της TCA δεν είχε πρακτικά επίδραση, επειδή ο released code είχε hard-coded τα layers 10–15 σε πέντε διαφορετικά σημεία. Ο κώδικας διορθώθηκε ώστε η παράμετρος να ελέγχει πραγματικά το εύρος και δοκιμάστηκαν τρεις εναλλακτικές αφετηρίες, στα layers 4, 8 και 12.

- **Πείραμα 5: guidance με object-specific prompt.** Η guidance scale επίσης δεν είχε επίδραση στη released configuration, επειδή το prompt ήταν κενό και οι conditional και unconditional predictions συνέπιπταν. Για να ελεγχθεί αν η ρητή ταυτότητα του αντικειμένου βοηθά, χρησιμοποιήθηκε ως prompt το object label του GeoBench και δοκιμάστηκαν guidance scales $w = 7{,}5$ και 10.

- **Πείραμα 6: TCA floor.** Το βάρος της TCA μειώνεται σταδιακά προς ένα κατώτατο όριο, το οποίο στη δημοσιευμένη ρύθμιση είναι 0. Δοκιμάστηκαν floors 0,25 και 0,5, ώστε η μεταφορά από το

Πίνακας 1.3: Οι έξι υποθέσεις της εργασίας, οι παρεμβάσεις που δοκιμάστηκαν και το συμπέρασμα κάθε μελέτης.

Υπόθεση	Τι δοκιμάστηκε	Συμπέρασμα
Υ1: καλύτερο tuning της υπάρχουσας μεθόδου (Πειρ. 3–7)	refinement strength, TCA layer range, object-prompt guidance, TCA floor, anti-aliased coarse composite	Απορρίπτεται
Υ2: λείπει η ταυτότητα του αντικειμένου (Πειρ. 8)	crop του αντικειμένου ως IP-Adapter image tokens, σε τέσσερα scales	Απορρίπτεται
Υ3: λείπει η γεωμετρία του edit (Πειρ. 9–11)	μετασχηματισμένη εικόνα και edge representation μέσω IP-Adapter, Canny geometry μέσω ControlNet και τέσσερις geometry-aware παρεμβάσεις μέσα στην TCA	Απορρίπτεται
Υ4: το refinement χάνει χρήσιμη πληροφορία του coarse composite (Πειρ. 12–17)	latent anchoring, coarse composite ως output, post-refinement interior restoration με διαφορετικά widths και scope, πρώτα routed candidates	Υποστηρίζεται επιλεκτικά
Υ5: τα severe resizes χρειάζονται καλύτερα καθοδηγούμενη ανακατασκευή (Πειρ. 18–21)	projected guidance σε ε-space και D-space, frequency-decoupled guidance και διαφορετικά scopes εφαρμογής	Υποστηρίζεται για severe resizes
Υ6: η δομή του composite πρέπει να εισαχθεί μέσα στο αντικείμενο (Πειρ. 22–23)	spectral feature fusion σε διαφορετικές frequency bands, με διαφορετικό reference timing και με ή χωρίς guidance	Δεν υποστηρίζεται πέρα από το EPSREC

reference branch να μην εξαφανίζεται πλήρως, καθώς και ένα difficulty-aware floor που αυξάνεται με τη δυσκολία του edit.

- **Πείραμα 7: anti-aliased coarse composite.** Το αρχικό bilinear warp αντικαταστάθηκε από pipeline που πρώτα κάνει 2× upsampling με Lanczos, εφαρμόζει τον affine transform στην υψηλότερη ανάλυση και επιστρέφει στην αρχική με area averaging. Στόχος ήταν να περιοριστούν aliasing artifacts, κυρίως στα downscales. Η αλλαγή δοκιμάστηκε μόνη της και σε συνδυασμό με το TCA floor του Πειράματος 6.

Καμία από αυτές τις αλλαγές δεν αύξησε τη SUBC σε δύσκολη υποομάδα περισσότερο από 0,0023. Από αυτό το εύρος μεταβολών προέκυψε το πρακτικό threshold +0,005 για τους επόμενους μηχανισμούς. Η μόνη ένδειξη που άξιζε να μεταφερθεί παρακάτω ήταν το object-specific prompt, το οποίο μείωσε τη MD στα δύσκολα resizes κατά 1,73 pixels και επανεξετάστηκε αργότερα μέσα στη guidance family.

1.4.2 Υ2 και Υ3: πληροφορία που λείπει από το μοντέλο (Πειράματα 8–11)

Το refinement δεν λαμβάνει ποτέ ρητά ούτε την ταυτότητα του αντικειμένου ούτε τις παραμέτρους του γεωμετρικού μετασχηματισμού. Ένα frozen U-Net μπορεί να δεχθεί πρόσθετη πληροφορία από τρία βασικά entry points: μέσω image tokens στο cross-attention, μέσω spatial feature maps στα skip connec-

tions και μέσω παρεμβάσεων στο self-attention. Τα Πειράματα 8–11 καλύπτουν και τις τρεις δυνατότητες.

- **Πείραμα 8: ταυτότητα μέσω image tokens.** Το IP-Adapter κωδικοποιεί μια εικόνα με CLIP image encoder και εισάγει την πληροφορία της σε κάθε cross-attention layer μέσω ανεξάρτητου attention branch. Ως image prompt χρησιμοποιήθηκε crop του αρχικού αντικειμένου πάνω σε γκρι background. Η επίδραση περιορίστηκε στα edit branches και μέσα στη target mask, και δοκιμάστηκαν τέσσερα scales: $\lambda = 0{,}5$, 0,8, 1,1 και 1,4.

- **Πείραμα 9: γεωμετρία μέσω image tokens.** Με τον ίδιο μηχανισμό του IP-Adapter, αντί για το αρχικό crop χρησιμοποιήθηκε είτε το crop μετασχηματισμένο στη θέση-στόχο είτε συνδυασμός του αρχικού crop με ένα edge representation της μετασχηματισμένης εκδοχής, βασισμένο στο Sobel gradient magnitude. Δοκιμάστηκαν βάρη 0,5 και 1.

- **Πείραμα 10: γεωμετρία μέσω residual feature maps.** Το ControlNet εισάγει spatially aligned feature maps στα skip connections του frozen U-Net. Ως condition χρησιμοποιήθηκαν οι Canny edges του coarse composite γύρω από τη target mask, με scale 0,9 και μόνο στα edit branches. Δοκιμάστηκε μόνο του, μαζί με IP-Adapter και μαζί με IP-Adapter και object-specific prompt.

- **Πείραμα 11: γεωμετρία μέσα στην TCA.** Η γνωστή affine γεωμετρία ενσωματώθηκε απευθείας στον attention transplant με τέσσερις διαφορετικούς τρόπους: correspondence bias προς τη γεωμετρικά αντίστοιχη θέση της πηγής (G2, σε δύο εντάσεις), warped source keys/values στη θέση-στόχο (G1), gating που μηδενίζει τη μεταφορά όπου δεν υπάρχει έγκυρη αντιστοίχιση (G3), και reference branch που προκύπτει από την ήδη μετασχηματισμένη εικόνα της πηγής (R^2-TCA). Κατά την υλοποίηση διαπιστώθηκε επίσης ότι ο released code εφάρμοζε τη μάσκα του αντικειμένου μόνο στα attention heads 0, 2, 4 και 6 κάθε branch· για δίκαιη σύγκριση, όλοι οι μηχανισμοί αξιολογήθηκαν απέναντι σε διορθωμένο per-branch head layout.

Κανένα από τα παραπάνω entry points δεν βελτίωσε τη subject consistency στις δύσκολες ομάδες. Αντίθετα, όσο πιο άμεσα επιχειρούσε ο μηχανισμός να επιβάλει τη γεωμετρία του αντικειμένου μέσα στο denoising, τόσο περισσότερο έτεινε να μειώνει τη συνέπειά του.

1.4.3 Υ4: αγκύρωση στο coarse composite (Πειράματα 12–17)

Το coarse composite περιέχει ήδη το αντικείμενο στη σωστή θέση και με τον ζητούμενο affine transform. Η επόμενη υπόθεση είναι επομένως ότι το refinement καταστρέφει μέρος αυτής της χρήσιμης deterministic πληροφορίας και ότι η τελική έξοδος πρέπει, όπου είναι ασφαλές, να παραμένει πιο κοντά στο composite.

- **Πείραμα 12: latent anchoring κατά το denoising.** Μετά από κάθε refinement step, το edit latent αναμειγνύεται, στη ζώνη γύρω από το αντικείμενο, με το latent της αντίστοιχης coarse-composite trajectory, με βάρος w από 0,3 έως 0,9.

- **Πείραμα 13: το coarse composite ως τελική έξοδος.** Το composite αξιολογείται απευθείας, χωρίς refinement, τόσο στο development όσο και στο held-out subset. Ο στόχος είναι να απομονωθεί τι προσθέτει και τι αφαιρεί το diffusion refinement σε σχέση με τη deterministic λύση.

- **Πείραμα 14: interior restoration και πρώτα routed candidates.** Μετά την ολοκλήρωση του refinement, ένα eroded εσωτερικό τμήμα του αντικειμένου αντικαθίσταται από τα αντίστοιχα pixels του coarse composite, ενώ ένα boundary band παραμένει όπως το παρήγαγε το FreeFine. Ο μηχανισμός συνδυάστηκε με transform-specific επιλογή συμπεριφοράς σε τρία πρώτα routed models, τα οποία αξιολογήθηκαν και στο held-out subset.

- **Πείραμα 15: πλάτος του refined boundary band.** Στις μετατοπίσεις δοκιμάστηκαν widths $w = 4$, 8, 16 και 24 pixels, ώστε να μετρηθεί πόσο μεγάλο μέρος του deterministic interior είναι ασφαλές να διατηρηθεί.

- **Πείραμα 16: γιατί λειτουργεί το interior restoration.** Ο ίδιος post-refinement μηχανισμός εφαρμόστηκε είτε μόνο στα moves είτε global, και το coarse composite των μετατοπίσεων επανυπολογίστηκε τόσο με exact integer shift όσο και με anti-aliased warp. Με αυτόν τον τρόπο ελέγχθηκε αν το gain προέρχεται από το scope του preservation ή απλώς από καλύτερο interpolation.

- **Πείραμα 17: interior restoration ανά transform.** Ο μηχανισμός αξιολογήθηκε χωριστά σε moves ($w = 4$), rotations και resizes ($w = 8$), με τα resizes να αναλύονται επιπλέον στο δύσκολο τριτημόριο και στα υπόλοιπα δύο.

Το interior restoration βελτίωσε τη subject consistency στις μετατοπίσεις και στα ηπιότερα resizes, έβλαψε τις περιστροφές και δεν πρόσφερε ουσιαστικό όφελος στα δύσκολα resizes. Το αποτέλεσμα αυτό ήταν το πρώτο καθαρό σήμα ότι η σωστή στρατηγική πρέπει να εξαρτάται από τον τύπο και την ένταση του μετασχηματισμού.

1.4.4 Υ5: guidance για severe resizes (Πειράματα 18–21)

Στα severe resizes το αντικείμενο πρέπει να ανακατασκευαστεί σε κλίμακα που η πηγή δεν παρέχει άμεσα. Μέχρι εκείνο το σημείο, η μόνη αλλαγή που είχε δώσει χρήσιμο σήμα σε αυτή την κατηγορία ήταν το object-specific prompt. Τα Πειράματα 18–21 εξετάζουν ποια μορφή guidance μπορεί να αξιοποιήσει καλύτερα αυτή την πληροφορία.

- **Πείραμα 18: Adaptive Projected Guidance (APG) στο ε-space.** Η APG διαχωρίζει τη guidance correction σε παράλληλη και ορθογώνια συνιστώσα ως προς την conditional prediction, εφαρμόζει norm control και χρησιμοποιεί negative momentum ώστε να αποσβένει επαναλαμβανόμενες ωθήσεις στην ίδια κατεύθυνση. Ο μηχανισμός εφαρμόστηκε στις noise predictions, τόσο με το κανονικό όσο και με συντομότερο, ασθενέστερο refinement window.

- **Πείραμα 19: Frequency-Decoupled Guidance (FDG).** Η guidance update χωρίστηκε με 5×5 low-pass φίλτρο σε low- και high-frequency συνιστώσα. Οι δύο ζώνες έλαβαν διαφορετικά guidance scales: $w = 10$ για τις χαμηλές συχνότητες και 5 ή 15 για τις υψηλές, ώστε να ελεγχθεί αν η δομή και η λεπτομέρεια χρειάζονται διαφορετική ενίσχυση.

- **Πείραμα 20: APG στο clean-sample D-space και διαμόρφωση του EPSREC.** Οι ίδιες βασικές πράξεις μεταφέρθηκαν στην predicted clean sample representation, όπου σχεδιάστηκε αρχικά η APG. Ελεγχόμενες παραλλαγές απομόνωσαν το norm threshold, το projection, την αναδρομική momentum και το πού αποθηκεύεται η temporal state. Από αυτή τη μελέτη προέκυψε το EPSREC, το οποίο κρατά τη χρονική συσσώρευση στο ε-space και τη μετατρέπει στο D-space μόνο με τον scale

factor του τρέχοντος timestep.

- **Πείραμα 21: scope της guidance.** Αξιολογήθηκαν διαφορετικές πολιτικές εφαρμογής σε όλες τις 200 development περιπτώσεις: απλή object-prompt guidance global, EPSREC global, EPSREC μέσα σε routed model, EPSREC μόνο στα δύσκολα resizes και, ως control, EPSREC χωρίς object-specific prompt.

Στο development subset, το EPSREC μείωσε τη MD των δύσκολων resizes από 19,47 σε 16,61 pixels. Στα υπόλοιπα edit types, η ίδια guidance δεν έδωσε σταθερό όφελος και συχνά ήταν επιβαρυντική. Το αποτέλεσμα επιβεβαιώνει ότι ο μηχανισμός είναι specialist για τις έντονες αλλαγές κλίμακας και όχι γενική αντικατάσταση της guidance του FreeFine.

1.4.5 Υ6: δομή του composite μέσα στο αντικείμενο (Πειράματα 22–23)

Η local guidance του FreeFine δρα σε ζώνη γύρω από το αντικείμενο και όχι μέσα στο εσωτερικό του. Η τελευταία υπόθεση εξετάζει αν το refinement χρειάζεται επιπλέον structural information από το coarse composite απευθείας μέσα στα internal features του αντικειμένου.

- **Πείραμα 22: spectral feature fusion.** Σε ένα fixed decoder feature map του U-Net, ανάλυσης 32×32 με 1.280 channels, το edit feature αναμειγνύεται μέσα στο foreground με το αντίστοιχο reference/coarse feature μόνο σε επιλεγμένες spatial frequency bands. Η injection strength μειώνεται στα πρώτα refinement steps. Δοκιμάστηκαν high-frequency injection με τρεις εντάσεις, full-band injection, μικρότερο temporal horizon και εφαρμογή στο rotation branch.

- **Πείραμα 23: generalized spectral search.** Εξετάστηκαν πέντε spectral selectors (full, high, mid, mid+high και direction-aware high frequencies), καθώς και same-step έναντι fixed reference timing. Οι παραλλαγές αξιολογήθηκαν τόσο με όσο και χωρίς guidance και σε συνδυασμό με EPSREC, πάνω σε ολόκληρο το development subset.

Το spectral feature fusion μείωσε τη MD στα δύσκολα resizes μόνο όταν συνδυάστηκε με guidance· χωρίς guidance την αύξησε. Πάνω στο EPSREC, η πρόσθετη συνεισφορά του δεν ξεχώριζε από τη διακύμανση της MD. Η Υ6, όπως διατυπώθηκε, δεν υποστηρίζεται επομένως πέρα από το EPSREC· παρατηρήθηκε μόνο μια αλληλεπίδραση με τη guidance, και το spectral feature fusion δεν διατηρήθηκε ως ανεξάρτητο component της τελικής μεθόδου.

1.4.6 Το κοινό μοτίβο

Τα πειράματα αποκαλύπτουν ένα επαναλαμβανόμενο μοτίβο. Οι μηχανισμοί που επιχειρούν να επιβάλουν άμεσα τη δομή του αντικειμένου μέσα στο denoising τείνουν να το βλάπτουν. Αντίθετα, βοηθούν είτε μηχανισμοί που κατευθύνουν την ανακατασκευή γύρω από το αντικείμενο, όπως η guidance στα severe resizes, είτε μηχανισμοί που διατηρούν μετά το refinement ένα τμήμα του coarse composite που έχει αποδειχθεί αξιόπιστο. Εξίσου σημαντικό είναι ότι κάθε παρέμβαση που βοηθά μία κατηγορία edits μπορεί να βλάψει κάποια άλλη όταν εφαρμόζεται global. Επομένως, η τελική μέθοδος δεν πρέπει να επιλέξει έναν καθολικό μηχανισμό, αλλά να κάνει transform-aware routing.

1.5 Η προτεινόμενη μέθοδος

1.5.1 Αποκατάσταση του εσωτερικού

Μετά την ολοκλήρωση του refinement, το εσωτερικό του αντικειμένου αποκαθίσταται από το coarse composite, ενώ ένα boundary band πλάτους w pixels παραμένει όπως το παρήγαγε το FreeFine:

$$\hat{I} = M_{\text{int}} \odot I_c + (1 - M_{\text{int}}) \odot \hat{I}_{\text{FF}}, \qquad M_{\text{int}} = M_t \ominus B_{2w+1}, \tag{1.4}$$

όπου $\ominus$ δηλώνει morphological erosion με τετράγωνο structuring element διαστάσεων $(2w+1)\times(2w+1)$. Με αυτόν τον τρόπο, το εσωτερικό κρατά τη γεωμετρική ακρίβεια της deterministic σύνθεσης, ενώ το boundary παραμένει διαθέσιμο στο diffusion refinement ώστε να διορθωθούν seams και να γίνει φυσικό blending με το background.

1.5.2 EPSREC

Σε κάθε refinement step i, το object-specific prompt παράγει guidance correction $\Delta\varepsilon_i = \varepsilon_p - \varepsilon_\varnothing$. Το EPSREC δεν εφαρμόζει απλώς αυτή τη διαφορά με σταθερό scale. Διατηρεί μια temporal state u_i στο ε-space, τη συσσωρεύει αναδρομικά με αρνητικό momentum και στη συνέχεια τη μετατρέπει, μόνο για το τρέχον timestep, στο clean-sample D-space μέσω του συντελεστή $c_i = \sqrt{(1-\bar{\alpha}_{t_i})/\bar{\alpha}_{t_i}}$. Εκεί η correction αποσυντίθεται ως προς την conditional prediction D_p και αφαιρείται η παράλληλη συνιστώσα πριν ανακατασκευαστεί η guided prediction:

$$u_i = \Delta\varepsilon_i + \mu\, u_{i-1}, \qquad d_i = -c_i\, u_i, \qquad D^g = D_p + (w-1)(d_i - \mathrm{P}_{D_p} d_i), \tag{1.5}$$

με $\mu = -0{,}75$ και guidance scale $w = 10$. Το μέτρο της temporal state περιορίζεται στο 5 και η διορθωμένη D-space prediction μετατρέπεται ξανά σε noise prediction μέσω της Εξ. (1.2). Το αρνητικό momentum αποσβένει επαναλαμβανόμενη κίνηση προς την ίδια κατεύθυνση, ενώ η αφαίρεση της παράλληλης συνιστώσας περιορίζει την υπερενίσχυση μιας κατεύθυνσης που ήδη υπάρχει στην conditional prediction. Όπως και στο FreeFine, η guidance εφαρμόζεται μόνο στη local περιοχή γύρω από το αντικείμενο.

1.5.3 Ο κανόνας routing και το SGR-EPSREC

Ο τύπος του edit προκύπτει απευθείας από τις ζητούμενες affine παραμέτρους. Για τα resizes ο router χρησιμοποιεί την effective scale $s = \sqrt{s_x s_y}$:

$$\text{έντονη}(s) \quad \Longleftrightarrow \quad s \geq 1{,}5 \;\; \text{ή} \;\; s \leq 0{,}6, \tag{1.6}$$

όπου τα κατώφλια αντιστοιχούν στον ορισμό που χρησιμοποιούν οι δημιουργοί του GeoBench για μια δύσκολη αλλαγή κλίμακας και δεν επιλέχθηκαν με βάση κάποια μετρική.

Το SGR-EPSREC (Selective Geometric Refinement) κάνει routing κάθε edit στον χειρισμό που υποστηρίζεται από τα αντίστοιχα πειράματα του Πίνακα 1.4. Δεν εκπαιδεύει καμία νέα παράμετρο και δεν χρησιμοποιεί learned router· διαβάζει μόνο το requested transform. Όταν οι πρόσθετες παρεμβάσεις απενεργοποιούνται, το pipeline επιστρέφει στο reproduced FreeFine και παράγει byte-identical outputs. Στο development subset, κάθε branch συμπεριφέρθηκε σύμφωνα με το mechanistic signal που είχε καθιερωθεί στα προηγούμενα πειράματα (Πείραμα 25).

Πίνακας 1.4: Οι τέσσερις branches του SGR-EPSREC και τα πειράματα που στηρίζουν κάθε επιλογή.

Edit	Inference-time χειρισμός	Πειράματα
Μετατόπιση	FreeFine refinement + interior restoration, $w = 4$	15, 17
Περιστροφή	Original FreeFine, χωρίς πρόσθετη παρέμβαση	8–11, 17, 21–23
Non-severe resize	FreeFine refinement + interior restoration, $w = 8$	17
Severe resize	Object-specific prompt + CFG 10 + EPSREC guidance rectification	20, 21

Πίνακας 1.5: FreeFine και SGR-EPSREC στο πλήρες GeoBench-2D, σε όλες τις 5.677 επεξεργασίες.

Μετρική	FreeFine	SGR-EPSREC	Μεταβολή
FID↓	35,04	34,78	−0,26
FID_{dino}↓	487,82	487,36	−0,47
KD↓	0,1433	0,1417	−0,0016
SUBC↑	0,9113	0,9144	+0,0031
BGC↑	0,9670	0,9671	+0,0001
WRAP_E↓	0,0474	0,0292	−38,4%
MD↓	8,50	8,18	−3,8%

1.6 Αποτελέσματα στο πλήρες σύνολο

Το SGR-EPSREC αξιολογήθηκε σε όλες τις 5.677 επεξεργασίες του GeoBench-2D (Πείραμα 26) απέναντι σε ένα fresh FreeFine baseline που παρήχθη στο ίδιο evaluation pipeline και είναι πανομοιότυπο με την προηγούμενη αναπαραγωγή. Ο Πίνακας 1.5 συνοψίζει το κύριο αποτέλεσμα: σε aggregate επίπεδο, και οι επτά μετρικές κινούνται αριθμητικά προς την ευνοϊκή κατεύθυνση σε σχέση με το reproduced baseline.

Η ανάλυση ανά branch επιβεβαιώνει το μοτίβο που είχε παρατηρηθεί στο development subset. Στις μετατοπίσεις η SUBC αυξάνεται κατά 0,0073, με trade-off περίπου 0,19 pixel στη MD. Στα non-severe resizes η SUBC αυξάνεται κατά 0,0035, ενώ στα severe resizes η MD μειώνεται κατά περίπου 0,34 pixel. Οι περιστροφές παραμένουν ακριβώς ίδιες με το FreeFine, επειδή ο router επιλέγει συνειδητά το untouched baseline branch. Παρά το γεγονός ότι το coarse composite επανεισάγεται σε περισσότερα από 3.300 edits μέσω interior restoration, οι distributional realism metrics δεν υποβαθμίζονται.

Επειδή το baseline και η τελική μέθοδος αξιολογούνται πάνω στα ίδια edits, είναι δυνατή και paired ανάλυση ανά edit. Η SUBC αυξάνεται κατά μέσο όρο κατά 0,0031, με 95% confidence interval από 0,0029 έως 0,0032. Η βελτίωση δεν περιορίζεται σε λίγες μεμονωμένες περιπτώσεις: περίπου το 54% των edits βελτιώνεται, το 18% χειροτερεύει και το υπόλοιπο αντιστοιχεί κυρίως στις αμετάβλητες περιστροφές. Το αποτέλεσμα παραμένει πρακτικά ίδιο όταν εξαιρούνται τα edits που χρησιμοποιήθηκαν στο development.

Τέλος, η εργασία ξεχωρίζει ποιο μέρος των aggregate gains αντιστοιχεί σε ουσιαστικά καλύτερο editing και ποιο μέρος ευνοείται από τον ίδιο τον ορισμό των metrics. Η μείωση της WRAP_E είναι σε μεγάλο βαθμό αναμενόμενη από κατασκευή, επειδή το interior restoration αντιγράφει ακριβώς pixels από το coarse composite, δηλαδή από την ίδια αναφορά με την οποία συγκρίνει η μετρική. Κάτι ανάλογο ισχύει εν μέρει και για τη SUBC. Η πολύ μικρή άνοδος της BGC συνδέεται με λεπτομέρεια της implementation της μετρικής, η οποία μετρά ως background ένα λεπτό όριο pixels του αντικειμένου, κατά μήκος του

περιγράμματός του στην εικόνα πηγής. Αντίστοιχα, η aggregate MD επηρεάζεται από το διαφορετικό πλήθος keypoints που συνεισφέρει κάθε edit type. Το πιο άμεσο γεωμετρικό αποτέλεσμα είναι επομένως η μείωση της MD κατά περίπου 0,34 pixel στα severe resizes, μαζί με το γνωστό trade-off περίπου +0,19 pixel στις μετατοπίσεις.

1.7 Συμπεράσματα

Η εργασία δείχνει ότι, στο μεγαλύτερο μέρος του GeoBench-2D, το FreeFine βρίσκεται ήδη κοντά σε αυτό που επιτρέπει η deterministic πληροφορία της εισόδου του. Το υπόλοιπο περιθώριο βελτίωσης δεν κερδίζεται με μία ενιαία αλλαγή σε ολόκληρο το pipeline, αλλά με διαφορετικό χειρισμό ανά μετασχηματισμό, ανάλογα με το πόσο αξιόπιστο είναι το coarse composite. Όταν το deterministic interior είναι αξιόπιστο, συμφέρει να διατηρηθεί· όταν το αρχικό FreeFine διορθώνει καλύτερα τα artifacts, το baseline refinement πρέπει να μείνει αμετάβλητο· και όταν ο μετασχηματισμός είναι αρκετά έντονος ώστε να απαιτεί πραγματική ανακατασκευή, χρειάζεται πιο στοχευμένη guidance. Πάνω σε αυτή τη διάκριση στηρίζεται το SGR-EPSREC, το οποίο βελτιώνει αριθμητικά και τις επτά aggregate μετρικές του benchmark χωρίς επιπλέον training.

Τα όρια της εργασίας παραμένουν σαφή. Καμία από τις παρεμβάσεις που εξετάστηκαν δεν έδωσε αρκετά robust βελτίωση στις περιστροφές ώστε να αντικαταστήσει το FreeFine. Στις μετατοπίσεις, το κέρδος στη συνέπεια του αντικειμένου συνοδεύεται από μικρό trade-off στη MD. Στα severe resizes, η μείωση του keypoint error είναι μικρή σε απόλυτους όρους. Επιπλέον, όλα τα αποτελέσματα προέρχονται από ένα benchmark, από ένα random seed ανά edit και από τις διαθέσιμες automatic metrics, χωρίς human evaluation.

Οι φυσικές επεκτάσεις της εργασίας βρίσκονται πέρα από αυτά τα όρια. Οι περιστροφές πιθανόν να χρειάζονται παρέμβαση στο ίδιο το μοντέλο ή ένα μικρό trainable component ειδικά για rotation. Μια γεωμετρική μετρική που θα είναι λιγότερο ευαίσθητη στην appearance και θα έχει επικυρωθεί με ανθρώπινη αξιολόγηση θα επέτρεπε καθαρότερες συγκρίσεις. Παράλληλα, τα out-of-plane edits του GeoBench, το background/scene editing γύρω από ένα σταθερό foreground object και η μετάβαση σε νεότερα diffusion backbones θέτουν ξανά το ίδιο βασικό ερώτημα: ποια πληροφορία πρέπει να διατηρηθεί και ποια πρέπει να αναπαραχθεί από το generative model.

Chapter 2

Introduction

2.1 Geometric image editing

Moving an object to another place in a photograph, turning it, or making it larger or smaller are among the most natural edits a person can ask for, and among the hardest for a generative model to perform. *Geometric image editing* asks for exactly this. Given a photograph, the mask of an object in it and a transformation (a translation, a rotation or a change of scale), the result must show the object at its new position, orientation or size, it must still be recognisably the same object, and the rest of the scene must remain the same scene. Edits of this kind are central to photo retouching, product imagery, layout design and visual content creation, wherever an image is almost right and one object needs to be somewhere else.

The task combines four requirements that pull against each other. The object must appear at its target pose with its appearance intact. The region it leaves behind must be filled with plausible background. The object must fit into its new surroundings, with no visible seam and with lighting and occlusions that agree with the scene. And parts that the source never showed must sometimes be created: an enlarged object needs detail at a scale at which it was never photographed, and a rotated one exposes outlines that were never visible. Copying the object's pixels to the target satisfies the first requirement and fails the third; generating the object anew satisfies the third and risks the first. Every method for geometric editing is a way of balancing these two.

Diffusion models [2, 8] changed what this balance can achieve. They generate images of a quality that earlier generative models did not reach, and their ability to complete, harmonise and regenerate parts of an image made them the basis of modern image editing. Their generative strength is also the difficulty: a model that redraws an object can redraw it differently, and a geometric edit is judged precisely on whether the object stayed the same.

2.2 Training-free editing on frozen diffusion models

Large text-to-image diffusion models such as Stable Diffusion [8] encode, in their frozen weights, a broad knowledge of how images look. A family of *training-free* methods exploits this knowledge without changing a single weight: they invert a real image into the model's noise space and steer its regeneration, by manipulating the model's attention, by guiding its predictions, or by controlling which parts of the image are regenerated [7, 15–17, 19]. Such methods need no training data and no fine-tuning, and they apply to any image the model can represent. For geometric editing they range

from drag-based editors [23, 24, 26] to methods that operate on explicit object transformations [31, 32].

FreeFine [34] is the strongest of them on geometric edits. It decomposes a geometric edit into three stages: it transforms the object, inpaints the region the object vacates, and then refines the coarse composite in which the transformed object stands on the inpainted background. Both diffusion stages run on a frozen Stable Diffusion 1.5, and refinement keeps the object faithful to its source by transplanting the source object's features into the edited image through an attention mechanism, Temporal Contextual Attention. Its authors also introduced GeoBench, a benchmark of geometric edits with exact transformation parameters and seven metrics that measure realism, consistency and geometric precision, and on it FreeFine outperforms eight earlier editing methods. It is the natural starting point for any attempt to improve training-free geometric editing.

Its published evaluation, however, consists of averages over all edits. What these averages do not show is where FreeFine fails, why it fails there, and whether the failure can be removed without the training that the method was designed to avoid.

2.3 Problem statement and research questions

This thesis studies FreeFine on the 5,677 two-dimensional edits of GeoBench, GeoBench-2D, and sets out to improve it without training, guided by an understanding of its failure rather than by trial and error. Three research questions structure the work.

RQ1. Can FreeFine's results be reproduced faithfully from its public code, and on which edits of GeoBench-2D does it fail?

RQ2. What causes the failure: the configuration of the method, information that the frozen model lacks, or the way refinement treats the coarse composite it receives?

RQ3. Can FreeFine be improved without training, and by how much, once the improvements that reflect better editing are separated from those that the metrics grant by construction?

Answering them requires more than proposing a mechanism and reporting a better average. It requires a baseline that can be trusted, a diagnosis that names the edits on which the method fails, explanations that can be tested and rejected one at a time under controlled conditions, and an evaluation that knows what its metrics measure. The thesis is built around these four elements.

2.4 Contributions

The thesis makes six contributions.

1. **A faithful, deterministic reproduction of FreeFine.** The public code was run on all 5,677 edits of GeoBench-2D. It reproduces the published realism and consistency to within 2%, scores better on geometric precision, and regenerates bit-identical images across runs, which makes every later comparison exact, image by image. The analysis of the released code also established how

its controls actually act: the start step sets the strength of refinement, guidance is inactive in the released configuration because the prompt is empty, the attention layers it modulates are fixed in the code, and the region in which guidance acts excludes the object itself (Chapters 5 and 9).

2. **A metric-driven diagnosis and a controlled protocol.** The failure of FreeFine is concentrated in hard rotations and hard resizings, a quarter of the benchmark, where subject consistency and geometric precision fall while background consistency and the warping error do not move. The diagnosis also identifies the properties of the official metric code that any comparison has to respect, and the protocol under which every later experiment runs: a stratified development subset, a disjoint held-out subset, a switch-off check that proves each modification exact when disabled, paired generation and seeded stochastic metrics (Chapters 7 and 8).

3. **A systematic test of six explanations of the failure.** Twenty-one controlled experiments test, in turn, the configuration of the method, the identity of the object, the geometry of the edit (supplied through every entry point of a frozen U-Net: image tokens, residual feature maps and self-attention), anchoring to the coarse composite, guidance, and spectral feature fusion (Chapters 9 to 13). None of the tested reconfigurations and none of the tested identity- or geometry-conditioning mechanisms explains the failure; the evidence points instead to how refinement treats the composite, and shows that the right treatment depends on the transformation. The experiments also expose a general pattern: mechanisms that constrain the object while it is denoised harm it, and mechanisms that steer its surroundings, or keep what is reliable after refinement, help.

4. **EPSREC, a guidance for severe resizings.** A reformulation of adaptive projected guidance with a recursive momentum kept in noise space, together with an analysis of why it works: the gain comes from the label and from a momentum that damps repeated pushes, not from the projection or the threshold. On the hard resizings of the development subset it lowers the keypoint error from 19.47 to 16.61 pixels, by 14.7%, to within 0.17 pixels of the coarse composite itself (Chapter 12). On all 759 severe resizings of the benchmark the reduction is 0.34 pixels, or 1.8% (Chapter 15).

5. **SGR-EPSREC, a training-free, transform-aware refinement policy.** Selective Geometric Refinement routes each edit, from its requested transformation alone, to the treatment that the experiments established for it: interior restoration for translations and moderate resizings, FreeFine unchanged for rotations, and EPSREC for severe resizings, whose threshold is the benchmark's own definition of a hard resizing (Chapter 14). On all 5,677 edits it improves all seven metrics of GeoBench-2D over FreeFine, raises subject consistency for 54% of the edits while lowering it for 18%, reduces the keypoint error of severe resizings, leaves every rotation exactly as FreeFine produces it, and does not lower realism (Chapter 15).

6. **An account of what the benchmark's metrics measure.** Scoring the coarse composite, the source object copied to its target pose, as an output shows that FreeFine already reaches 99% of the composite's subject consistency, so that most of the drop measured on hard edits belongs to the transformation rather than to the method. The thesis further shows which gains any method of this kind obtains by construction (the warping error, part of subject consistency, and, through a property of the released code, background consistency) and why the whole-benchmark keypoint error depends on how many keypoints each kind of edit contributes. On this basis it separates the improvements that reflect better editing from those that do not (Chapters 15 and 16).

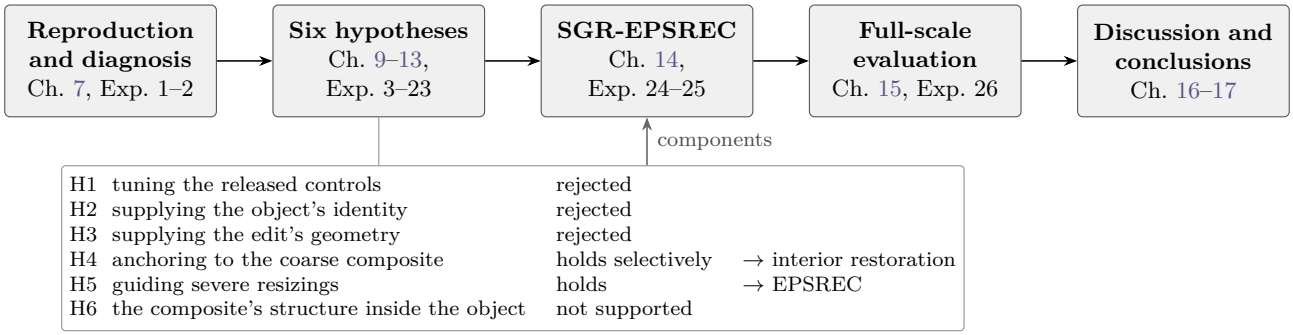


Figure 2.1: The structure of the thesis. The diagnosis of Chapter 7 defines the target; six hypotheses about the cause of the failure are tested in Chapters 9 to 13; the two that hold supply the components of SGR-EPSREC, which is evaluated on the whole benchmark.

All code, executed notebooks and results of the thesis are publicly available.[1]

2.5 Scope and non-goals

The thesis works with FreeFine's pipeline on a frozen Stable Diffusion 1.5 and changes only how it is run: no network is trained or fine-tuned, and every component it adds is computed at inference time. It studies the in-plane edits of GeoBench-2D (translations, rotations and resizings) and leaves the three-dimensional and structure-completion subsets of GeoBench aside. It measures with the benchmark's official metrics, which it analyses but does not replace. Its aim is not a new editing paradigm but a precise understanding of the strongest existing one, and an improvement that follows from that understanding.

2.6 Structure of the thesis

Figure 2.1 shows how the thesis is organised. Chapter 1 is an extended summary in Greek. Chapters 3 to 5 provide the background: diffusion models and Stable Diffusion, with the sampling, inversion and guidance that the later chapters modify (Chapter 3); the work on training-free editing, geometric editing, guidance and the evaluation of geometric edits among which the thesis is placed (Chapter 4); and FreeFine and GeoBench-2D, including the seven metrics and the properties of their implementation (Chapter 5).

Chapter 6 states the methodology of the thesis: the problem, the stages of the study, the six hypotheses and how each is tested, the method to which the study leads, and the design of the evaluation. Chapter 7 reproduces FreeFine and diagnoses its failure, and Chapter 8 sets out the protocol shared by all later experiments. Chapters 9 to 13 test the six hypotheses: tuning the released pipeline (Chapter 9), supplying information the model lacks (Chapter 10), anchoring the refinement to the coarse composite (Chapter 11), guiding the reconstruction of severe resizings (Chapter 12), and supplying the composite's structure inside the object (Chapter 13). Chapter 14 combines the components that hold into SGR-EPSREC, and Chapter 15 evaluates it on all 5,677 edits. Chapter 16 draws the results together and states their limitations, and Chapter 17 concludes.

[1] https://github.com/giorgostzamouranis/Diploma-Thesis

Every experiment, from the first to the twenty-sixth, is numbered once and reported in the same way: what is changed and why, the results, and what they establish. Shared material (the metrics, the subsets and the controls) is defined once, in Chapters 5 and 8, and referred to thereafter.

Chapter 3

Diffusion Models and Stable Diffusion

Every method studied in this thesis runs on the same frozen, pretrained network, Stable Diffusion 1.5, and changes how that network is sampled rather than how it was trained. This chapter presents the theory on which those changes rest: the diffusion process and its training objective (§3.1), deterministic sampling with DDIM (§3.2), the inversion that maps a real image back to noise (§3.3), partial denoising, which turns a sampler into an editor (§3.4), the architecture of Stable Diffusion 1.5 (§3.5), the attention operations inside it (§3.6), and classifier-free guidance (§3.7). The notation, collected in Table 3.2 at the end of the chapter, is used throughout the thesis.

3.1 Denoising diffusion probabilistic models

Diffusion models [1–3] generate data by learning to reverse a process that gradually turns data into noise.

The forward process. Starting from a data sample $x_0 \sim q(x_0)$, the forward process adds Gaussian noise in T small steps,

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t;\, \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t \mathrm{I}\big), \qquad t = 1, \dots, T, \tag{3.1}$$

with a variance schedule $0 < \beta_1 < \cdots < \beta_T < 1$. Writing $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the composition of these Gaussian steps is again Gaussian, so the noisy sample at any step can be drawn directly from the clean one:

$$q(x_t \mid x_0) = \mathcal{N}\big(x_t;\, \sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t)\mathrm{I}\big), \qquad \text{that is,} \qquad x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I}). \tag{3.2}$$

The coefficient $\sqrt{\bar{\alpha}_t}$ weights the signal and $\sqrt{1-\bar{\alpha}_t}$ the noise. Their squares sum to one, so a sample of unit variance keeps unit variance at every step. The ratio

$$\mathrm{SNR}(t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \tag{3.3}$$

is the signal-to-noise ratio of step t. It decreases monotonically, and the schedule is chosen so that $\bar{\alpha}_T \approx 0$ and x_T is close to pure noise.

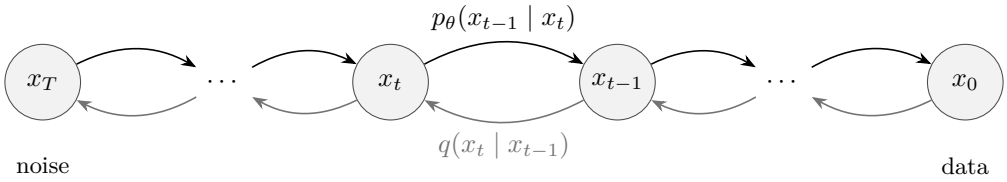


Figure 3.1: The forward process q adds Gaussian noise step by step (lower arrows, right to left); the learned reverse process p_θ removes it (upper arrows, left to right).

The reverse process. The reverse conditionals $q(x_{t-1} \mid x_t)$ depend on the unknown data distribution. Once the clean sample is also given, however, the reverse step is Gaussian with closed-form parameters:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1};\, \tilde{\mu}_t,\, \tilde{\beta}_t \mathrm{I}), \quad \tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\, x_0 + \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\, x_t, \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t. \tag{3.4}$$

A diffusion model learns a Gaussian $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 \mathrm{I})$ that approximates this step without knowing x_0. It generates by drawing $x_T \sim \mathcal{N}(0, \mathrm{I})$ and applying the learned step T times (Figure 3.1).

Predicting the noise. Solving Eq. (3.2) for x_0 and substituting into $\tilde{\mu}_t$ gives $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon\big)$: the reverse mean is known as soon as the noise that produced x_t is known. Ho et al. [2] therefore train a network $\varepsilon_\theta(x_t, t)$ to predict that noise, and set

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(x_t, t)\Big). \tag{3.5}$$

The model is trained by maximising a variational lower bound on the log-likelihood of the data. With this parameterisation every term of the bound is, up to a weight that depends on t, a squared error between the true and the predicted noise. Dropping the weights gives the objective used in practice,

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,\,\varepsilon,\,t}\Big[\big\|\varepsilon - \varepsilon_\theta\big(\sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon,\; t\big)\big\|^2\Big], \tag{3.6}$$

with t drawn uniformly from $\{1, \ldots, T\}$.

Two readings of the noise prediction. The noise prediction can be read in two further ways, and the later chapters use both.

- **A prediction of the clean sample.** Inverting Eq. (3.2) with the predicted noise gives the network's current estimate of the final result,

$$\hat{x}_0(x_t) = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{3.7}$$

  The minimiser of Eq. (3.6) is the conditional expectation $\mathbb{E}[\varepsilon \mid x_t]$, so $\hat{x}_0$ is the posterior mean of the clean sample: an average over all images consistent with x_t. It is blurry at high noise and becomes sharper as t decreases.

- **A scaled score.** The same minimiser equals, up to a factor, the gradient of the log-density of the noisy data distribution q_t [3, 4]:

$$\varepsilon^\star(x_t, t) = \mathbb{E}[\varepsilon \mid x_t] = -\sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t} \log q_t(x_t). \tag{3.8}$$

The noise prediction points away from regions of high probability. Guidance (§3.7) combines the scores of different distributions, and Eq. (3.8) is what allows it to do so by combining noise predictions.

3.2 Deterministic sampling with DDIM

Sampling with Eq. (3.5) requires all $T = 1000$ steps. Song et al. [5] observed that the objective of Eq. (3.6) depends only on the marginals of Eq. (3.2), not on the Markov chain that produced them. A whole family of non-Markovian processes shares those marginals, so a network trained once can be sampled with any member of the family. The reverse step of the family is

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{x}_0(x_t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\varepsilon_\theta(x_t,t) + \sigma_t\,\xi, \qquad \xi \sim \mathcal{N}(0,\mathrm{I}), \tag{3.9}$$

with

$$\sigma_t = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}. \tag{3.10}$$

The three terms are the predicted clean sample brought to the signal level of step $t-1$, the predicted noise brought to its noise level, and fresh noise. The parameter η selects the member of the family: $\eta = 1$ recovers the ancestral sampler of §3.1, and $\eta = 0$ removes the fresh noise and makes sampling deterministic. The deterministic sampler is known as DDIM, and η can be chosen at sampling time, even differently in different regions of the image.

Skipping steps. Because only the marginals matter, Eq. (3.9) remains valid when $t-1$ is replaced by any earlier timestep, so the sampler can run on a short increasing subsequence $t_1 < t_2 < \cdots < t_S$ of the T training timesteps. Stable Diffusion is usually sampled with $S = 50$. In the implementation used in this thesis the grid is $t \in \{1, 21, 41, \ldots, 981\}$, twenty timesteps apart, and sampling runs from $t = 981$ down to $t = 1$.

DDIM as an ODE solver. Dividing Eq. (3.9) with $\eta = 0$ by $\sqrt{\bar{\alpha}_{t-1}}$ reveals its structure. Define the rescaled sample $\bar{x}_t = x_t/\sqrt{\bar{\alpha}_t}$ and the noise-to-signal ratio

$$c_t = \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}} = \mathrm{SNR}(t)^{-1/2}. \tag{3.11}$$

By Eq. (3.7), $\hat{x}_0 = \bar{x}_t - c_t\,\varepsilon_\theta$, and the deterministic step between neighbouring grid points becomes

$$\bar{x}_{t_{k-1}} = \bar{x}_{t_k} + (c_{t_{k-1}} - c_{t_k})\,\varepsilon_\theta(x_{t_k}, t_k), \tag{3.12}$$

an Euler step of the ordinary differential equation $\mathrm{d}\bar{x} = \varepsilon_\theta\,\mathrm{d}c$ [5]. Two consequences are used later. First, DDIM defines a deterministic map between noise and images that can be run backwards (§3.3). Second, c_t is the exchange rate between the noise prediction and the predicted clean sample. By Eq. (3.7), a change $\Delta\varepsilon$ of the noise prediction changes the prediction of the final image by

$$\Delta\hat{x}_0 = -c_t\,\Delta\varepsilon. \tag{3.13}$$

Figure 3.2 shows how widely this rate varies over a sampling run. At the first step of the 50-step grid $c_{981} = 13.1$, and a small change of the noise prediction moves the predicted image a great deal; at the last step $c_1 = 0.041$, and the same change barely moves it.

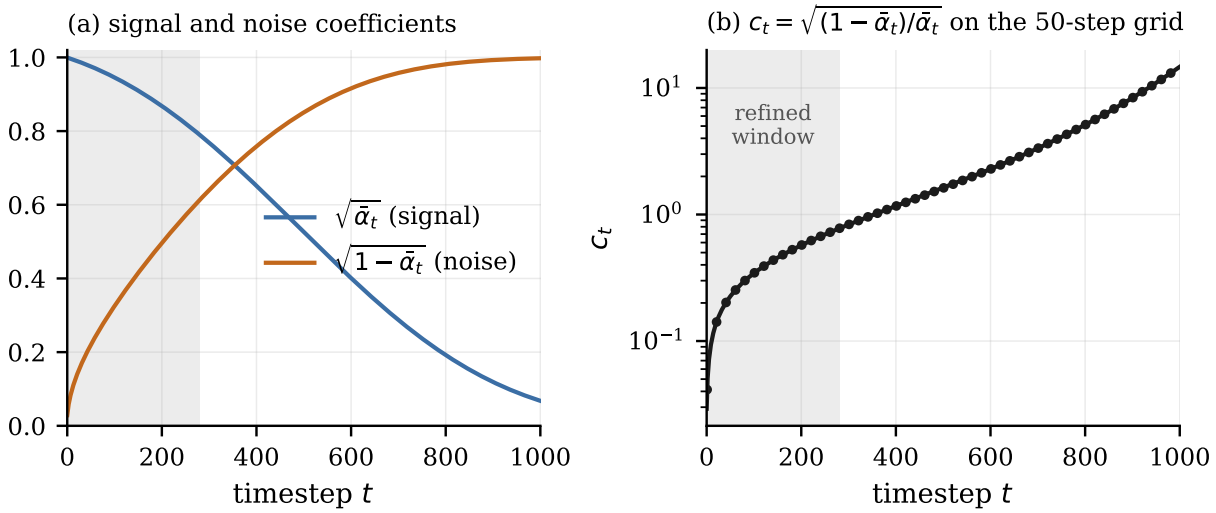


Figure 3.2: The noise schedule of Stable Diffusion 1.5 (scaled-linear β_t from 0.00085 to 0.012, $T = 1000$). (a) The signal and noise coefficients of Eq. (3.2). (b) The noise-to-signal ratio c_t of Eq. (3.11), on a logarithmic scale; the dots are the fifty timesteps of the DDIM grid. The shaded region is the window refined by the configuration of FreeFine studied in this thesis (§3.4).

3.3 DDIM inversion

Editing a real image requires the noise from which the sampler would produce that image. Running Eq. (3.12) backwards would require the noise prediction at the unknown next point, so DDIM inversion approximates it by the prediction at the current one:

$$\bar{x}_{t_{k+1}} = \bar{x}_{t_k} + (c_{t_{k+1}} - c_{t_k})\, \varepsilon_\theta(x_{t_k}, t_k). \tag{3.14}$$

Starting from the clean sample, repeated application yields an *inversion trajectory* $x_{t_1}, x_{t_2}, \ldots$ that ends in a noise-like latent, and running the sampler from that latent approximately reconstructs the image. The approximation assumes that the noise prediction changes little between neighbouring grid points. The error it introduces grows with the step size and is amplified by classifier-free guidance, which is why guided editing methods invest in correcting it, for example by optimising the unconditional text embedding at every step [6]. Without guidance the error is small in practice [5].

Two properties of inversion make it the basis of editing. The trajectory records the image at every noise level of the sampler, so it can be stored and used as a reference while a modified version is generated. And inversion need not run to the end of the grid: it can stop at any timestep, which is what partial denoising does.

3.4 Partial denoising and editing strength

SDEdit [7] turns a sampler into an editor. A guide image (a rough painting, a collage, or a photograph with an object inserted into it) is noised to an intermediate level t_0 with Eq. (3.2) and then denoised from t_0 to the end. The noise destroys detail below a certain scale while the coarse layout survives, and the sampler replaces the destroyed detail with content that is consistent with the model's image distribution. The entry level t_0 sets the editing strength.

- With a small t_0 only a few steps are run: the output stays close to the guide image, including its artefacts.

- With a large t_0 the output is realistic but departs from the guide image, and eventually loses its layout.

The entry level is therefore a trade-off between faithfulness to the input and realism of the output, and no single value is best for all inputs.

The deterministic variant replaces the random noise by DDIM inversion to t_0 (§3.3). The same input then always gives the same output, and when nothing is modified the sampler returns approximately the input itself; every difference between input and output is caused by whatever the editing method changes during the denoising steps. On the 50-step grid, choosing t_0 amounts to choosing how many of the fifty steps are run.

The configuration of FreeFine studied in this thesis (Chapter 5) enters at $t_0 = 281$ and runs the last fifteen steps of the grid, the shaded window of Figure 3.2. At that point $\sqrt{\bar{\alpha}_{281}} = 0.789$ and $\sqrt{1-\bar{\alpha}_{281}} = 0.615$, a signal-to-noise ratio of 1.65: the latent is still dominated by its input, so the layout of the image is fixed and refinement rewrites structure at the middle and fine scales. Across the window c_t falls from 0.779 to 0.041.

3.5 Latent diffusion and Stable Diffusion 1.5

Diffusion in pixel space is expensive at 512×512, because the denoising network must process 786,432 values at every step. Latent diffusion [8] moves the process into the latent space of a pretrained autoencoder. Stable Diffusion 1.5 (SD-1.5) is such a model. It has three components, shown in Figure 3.3.

The autoencoder. An encoder $\mathcal{E}$ maps an image $x \in \mathbb{R}^{512\times512\times3}$ to a latent $z = \mathcal{E}(x) \in \mathbb{R}^{64\times64\times4}$, a spatial reduction by a factor of eight, and a decoder $\mathcal{D}$ maps latents back to images. The autoencoder is trained with a reconstruction, a perceptual and a patch-based adversarial loss, together with a slight Kullback-Leibler penalty towards a standard Gaussian [8, 9], and its latents are multiplied by 0.18215 so that they have approximately unit variance. Every diffusion quantity of §3.1–§3.4 refers to this latent, which in the rest of the thesis is written z_t; the edited image is decoded only at the end.

The text encoder. A prompt is split into 77 tokens and encoded by the frozen text transformer of CLIP ViT-L/14 [10] into a sequence $c \in \mathbb{R}^{77\times768}$. The empty prompt is not a zero vector: it is tokenised into a start token, an end token and padding, and encodes to a fixed sequence $c_\varnothing$, which serves as the unconditional input of guidance (§3.7).

The denoising U-Net. The noise predictor $\varepsilon_\theta(z_t, t, c)$ is a U-Net [11] of about 860 million parameters [12] that works at four resolutions, 64×64, 32×32, 16×16 and 8×8, with 320, 640, 1280 and 1280 channels. The encoder applies two residual blocks per resolution, each followed at the three

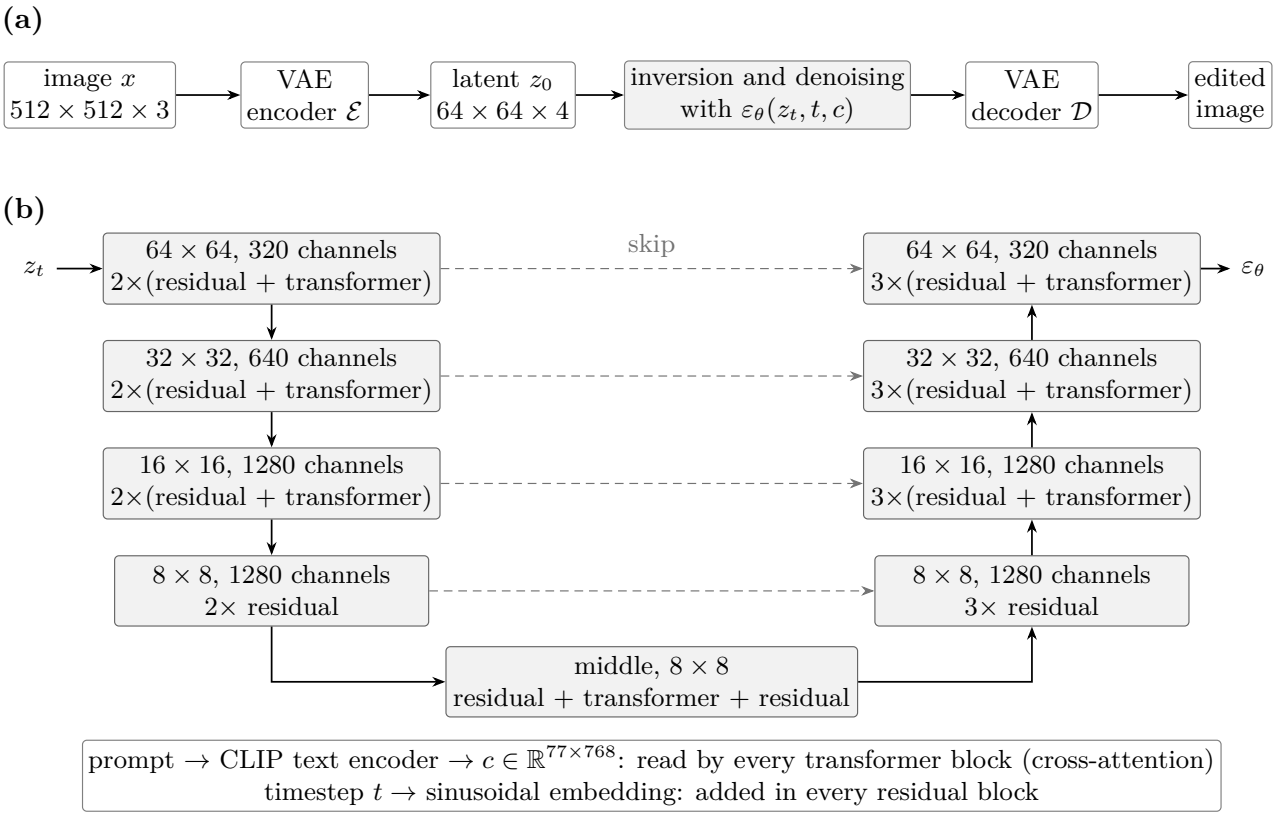


Figure 3.3: Stable Diffusion 1.5. (a) Diffusion runs in the latent space of an autoencoder: the image is encoded once, inverted and denoised as a latent, and decoded at the end. (b) The denoising U-Net ε_θ. Each box is one resolution level with its blocks; dashed arrows are the skip connections, which carry encoder features to the decoder.

finest resolutions by a transformer block, and halves the resolution between levels with a strided convolution. The middle block applies a residual block, a transformer block and a second residual block at 8×8. The decoder mirrors the encoder with three residual blocks per resolution, each followed at 16×16, 32×32 and 64×64 by a transformer block, and doubles the resolution between levels. Skip connections carry twelve feature maps of the encoder to the decoder, where they are concatenated with the decoder's features at the same resolution. The timestep enters through a sinusoidal embedding, passed through a small network and added inside every residual block. The U-Net contains sixteen transformer blocks in total: six in the encoder, one in the middle block and nine in the decoder. Table 3.1 lists them in the order in which they execute; later chapters address individual layers by this index.

The transformer block. Each transformer block turns its input feature map f, with C channels, into a sequence of tokens h, one per position, by group normalisation and a 1×1 convolution. It then applies three residual updates,

$$h \leftarrow h + \mathrm{SA}(\mathrm{LN}(h)), \qquad h \leftarrow h + \mathrm{CA}(\mathrm{LN}(h), c), \qquad h \leftarrow h + \mathrm{FF}(\mathrm{LN}(h)), \tag{3.15}$$

and adds the result back to f after a second 1×1 convolution. Here SA is self-attention and CA cross-attention (§3.6), LN is layer normalisation, and FF is a two-layer feed-forward network with a gated GELU activation and an inner width of $4C$ [13].

Table 3.1: The sixteen transformer blocks of the SD-1.5 U-Net, in execution order.

Index	Location	Resolution	Tokens	Channels	Head width
0–1	encoder	64×64	4096	320	40
2–3	encoder	32×32	1024	640	80
4–5	encoder	16×16	256	1280	160
6	middle	8×8	64	1280	160
7–9	decoder	16×16	256	1280	160
10–12	decoder	32×32	1024	640	80
13–15	decoder	64×64	4096	320	40

Every block has 8 attention heads, and every block cross-attends to the 77 text tokens of width 768.

Training. SD-1.5 was trained on image-text pairs from the LAION collection and fine-tuned at 512×512 with the text condition dropped for ten per cent of the samples, so that the same network provides both conditional and unconditional predictions for classifier-free guidance [12]. It is used frozen throughout this thesis.

3.6 Attention in the U-Net

Scaled dot-product attention. For queries $Q \in \mathbb{R}^{N \times d}$, keys $K \in \mathbb{R}^{M \times d}$ and values $V \in \mathbb{R}^{M \times d_v}$, attention [14] computes

$$\mathrm{Attn}(Q, K, V; B) = \mathrm{softmax}\Big(\frac{QK^\top}{\sqrt{d}} + B\Big)V, \tag{3.16}$$

where the softmax acts on each row and $B \in \mathbb{R}^{N \times M}$ is an optional additive bias, zero unless stated otherwise. Each output row is a convex combination of the values, weighted by the similarity of the query to each key. In multi-head attention the channels are split into H groups that attend separately and are concatenated afterwards. Every attention layer of SD-1.5 has $H = 8$ heads, with head widths of 40, 80 and 160 at 320, 640 and 1280 channels (Table 3.1).

Self-attention and cross-attention. In self-attention, queries, keys and values all come from the tokens of the same image, $Q = hW_Q$, $K = hW_K$, $V = hW_V$, so that every position gathers information from every other position; at 64×64 this is an attention over 4096 tokens. In cross-attention the queries come from the image and the keys and values from the conditioning, $Q = hW_Q$, $K = cW_K$, $V = cW_V$ with $M = 77$: every position reads from the prompt what is relevant to it.

Three properties that editing methods use.

1. **Masking through the bias.** Setting $B_{ij} = -\infty$ removes key j from the softmax of query i, $B_{ij} = 0$ leaves it untouched, and a finite negative value down-weights it. Restricting which keys a query may see, or which it should prefer, is a choice of B.

2. **No positional term.** The transformer blocks of SD-1.5 add no positional encoding, so the logits depend only on the features. Attention is therefore invariant to a joint permutation of the keys,

the values and the corresponding columns of the bias: for every permutation matrix P,

$$\mathrm{Attn}(Q, PK, PV; BP^\top) = \mathrm{Attn}(Q, K, V; B), \tag{3.17}$$

because permuting the columns of the logit matrix and the rows of V together permutes the terms of each weighted sum without changing it. Where a token lies in the image enters only through its features, which the convolutions of the residual blocks make position-dependent.

3. **Keys and values can come from another image.** Nothing requires Q, K and V to be computed from the same image. Using the queries of an image being generated with the keys and values of a reference image, known as mutual self-attention, makes the generated image take on the appearance of the reference wherever its queries match the reference's keys [15]. Editing methods exploit this and related manipulations of the attention layers [16, 17], and the attention transplant of FreeFine (Chapter 5) is of this kind.

3.7 Classifier-free guidance

Classifier guidance [18] steers sampling towards a condition by adding the gradient of a separately trained, noise-aware classifier to the score. Classifier-free guidance [19] obtains the same effect from the diffusion model alone. Because the network was trained both with and without the condition, it provides a conditional prediction $\varepsilon_c = \varepsilon_\theta(z_t, t, c)$ and an unconditional one $\varepsilon_\varnothing = \varepsilon_\theta(z_t, t, c_\varnothing)$, and guidance extrapolates from the second towards the first:

$$\tilde{\varepsilon} = \varepsilon_\varnothing + w\,(\varepsilon_c - \varepsilon_\varnothing). \tag{3.18}$$

Through Eq. (3.8) and Bayes' rule, $\log p(z_t \mid c) - \log p(z_t) = \log p(c \mid z_t) - \log p(c)$, the guided prediction is the scaled score of a sharpened distribution:

$$\tilde{\varepsilon} = -\sqrt{1-\bar{\alpha}_t}\,\nabla_{z_t}\big[\log p(z_t) + w \log p(c \mid z_t)\big]. \tag{3.19}$$

With $w = 1$ the sampler draws from the conditional model. With $w > 1$ it favours images that the implicit classifier $p(c \mid z_t)$ finds most typical of the condition, which improves the agreement with the prompt at the cost of diversity and, at large scales, of over-saturated colours. SD-1.5 is usually sampled with $w = 7.5$. Each guided step requires two predictions, which are computed together as a batch of two.

Three properties of guidance are used in later chapters.

- **Guidance needs a condition that differs from the unconditional input.** If $c = c_\varnothing$, then $\varepsilon_c = \varepsilon_\varnothing$ and $\tilde{\varepsilon} = \varepsilon_\varnothing$ for every w. With an empty prompt the sampler is unguided, whatever guidance scale is set.

- **Guidance is the same in noise space and in clean-sample space.** Since $\hat{x}_0$ is an affine function of the noise prediction with the same coefficients for both predictions, Eq. (3.7) gives $\hat{x}_0(\tilde{\varepsilon}) = \hat{x}_{0,\varnothing} + w\,(\hat{x}_{0,c} - \hat{x}_{0,\varnothing})$, and by Eq. (3.13) the guidance correction in clean-sample space is $-c_t$ times the correction in noise space. The two spaces differ only when the correction is transformed nonlinearly (normalised, projected or accumulated across steps), because c_t changes from one step to the next.

- **Guidance can be restricted to a region.** Multiplying the correction by a spatial mask m at latent resolution, $\tilde{\varepsilon} = \varepsilon_{\varnothing} + w\,(\varepsilon_c - \varepsilon_{\varnothing}) \odot m$, guides inside the mask and leaves the prediction unguided outside it.

Table 3.2: Notation introduced in this chapter and used throughout the thesis.

Symbol	Meaning
x_0, x_t; z_0, z_t	clean and noisy sample; in latent space
T	number of training timesteps (1000)
β_t, α_t, $\bar{\alpha}_t$	variance schedule, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s \le t} \alpha_s$
$\mathrm{SNR}(t)$, c_t	signal-to-noise ratio, Eq. (3.3); noise-to-signal ratio, Eq. (3.11)
ε, ε_θ	Gaussian noise; the network's noise prediction
$\hat{x}_0$	predicted clean sample, Eq. (3.7)
η	stochasticity of the sampling step, Eq. (3.10)
$t_1 < \cdots < t_S$	sampling grid; $S = 50$, $t \in \{1, 21, \ldots, 981\}$
t_0	entry timestep of partial denoising
c, $c_{\varnothing}$	text conditioning; conditioning of the empty prompt
w	guidance scale, Eq. (3.18)
Q, K, V, B	queries, keys, values and additive bias of attention, Eq. (3.16)
H	number of attention heads (8 in SD-1.5)

Chapter 4

Related Work

Geometric image editing asks a diffusion model to do something that most editing methods are built to avoid: to change *where* an object is while keeping *what* it looks like exactly. This chapter places the thesis among the work that addresses that task and the tools it draws on. It covers training-free editing in general (§4.1), drag- and point-based editing (§4.2), geometric and object-level editing (§4.3), the conditioning of frozen models (§4.4), guidance beyond classifier-free guidance (§4.5), the manipulation of the network's internal features (§4.6) and the evaluation of geometric editing (§4.7), and it ends by stating what distinguishes this thesis (§4.8). Methods that the later experiments build on are only introduced here; how they work and how they were integrated is explained in the experiment that uses them.

4.1 Training-free editing with diffusion models

The first diffusion editors modified the sampling of a pretrained model rather than its weights. SDEdit [7] noises a guide image to an intermediate level and denoises it (§3.4). DiffEdit [20] adds a mask: it contrasts the noise predictions of the model under two prompts to locate the region that an edit should change, and regenerates only that region after inverting the image. These methods decide how much of an image is regenerated, but not how its structure is carried into the result.

A second line of work carries the structure through the attention layers. Prompt-to-Prompt [16] edits an image by changing its prompt while keeping the cross-attention maps of the original prompt, so that the layout of the scene is preserved; for real images it relies on an accurate inversion, which null-text inversion [6] provides by optimising the unconditional text embedding at every step. Plug-and-Play [17] injects spatial features and self-attention maps of the source image into the generation of a new prompt. MasaCtrl [15] turns self-attention into *mutual* self-attention, in which the image being generated queries the keys and values of a source image, and uses masks derived from cross-attention to keep foreground and background apart. Cross-image attention [21] combines the queries of a structure image with the keys and values of an appearance image to transfer appearance without training. Trained editors such as InstructPix2Pix [22], which learns to follow written instructions from a synthetic dataset, form a separate family that this thesis does not pursue.

All of these methods keep the layout of the image and change its content or appearance; their attention injections exist to hold the structure in place. Geometric editing inverts that assumption. The appearance of the object must be kept exactly, and its position, orientation and size must change. Mutual self-attention remains the central tool, because it is how a training-free editor copies appearance from

the source; FreeFine's attention transplant (Chapter 5) is of this kind.

4.2 Drag-based and point-based editing

Drag-based editing lets a user move points of an image to new positions. DragGAN [23] introduced the interaction for generative adversarial networks: it optimises the latent code with a feature-based motion loss and tracks the dragged points as they move, but it is confined to the domains on which a GAN was trained.

DragDiffusion [24] brings the interaction to diffusion models. It optimises the diffusion latent at one timestep with supervision from the U-Net's features, fine-tunes a low-rank adaptation (LoRA) of the model on each input image to preserve its identity, and uses a latent variant of mutual self-attention. It also introduced DragBench, the first benchmark for point-based editing, with two metrics: an image fidelity score, one minus the LPIPS distance between the input and the edited image, and a *mean distance* that locates the dragged points in the edited image with DIFT correspondence [25] and measures how far they lie from their targets. The keypoint metric of GeoBench (§5.7) descends from this measure.

DragonDiffusion [26] removes the fine-tuning. It builds a classifier-guidance signal from the correspondence of intermediate diffusion features, applies it at several feature scales, and keeps the result consistent with the source through cross-branch self-attention, which lets it move and resize objects, replace their appearance and drag content without training. DiffEditor [27] extends it with image prompts, whose encoder is the only trained component, with a sampler that mixes stochastic and deterministic steps, and with regional score-based guidance and a time-travel strategy. RegionDrag [28] replaces points by regions: it copies the latents under a handle region to the corresponding positions of a target region at several timesteps and stabilises the result with mutual self-attention, completing an edit in one pass and more than a hundred times faster than DragDiffusion.

These methods express the edit as a displacement of points or regions, not as an exact transformation, and most of them realise it through optimisation or through gradient guidance at every step. A large rotation or rescaling of a whole object is not natural to express as a set of dragged points, and the guidance must discover the transformation that a geometric edit already states.

4.3 Geometric and object-level editing

A third family edits whole objects by an explicit geometric operation.

Guidance-based. Self-guidance [29] reads properties of an object (its position, size, shape and appearance) from the attention maps and activations of the model, and steers generation with gradients on those properties, so that an object can be moved or resized without training. Motion Guidance [30] lets the user specify a dense motion field and steers sampling with gradients through an off-the-shelf optical-flow network, which requires back-propagation through that network at every step.

Geometry inside the model. GeoDiffuser [31] treats an edit as a geometric transformation and applies it inside the attention layers of the diffusion model, with a zero-shot optimisation that preserves the style of the object and the photorealism of the scene; it handles translation and removal in two dimensions and rotation in three. Diffusion Handles [32] lifts the activations of an object to three dimensions with an estimated depth map, transforms depth and activations together, and projects them back, so that the regenerated image shows plausible occlusion and lighting without fine-tuning. DesignEdit [33] decomposes the latent into object layers and an incomplete background layer, fills the background with a key-masking variant of self-attention, and fuses the layers on a new canvas according to the instruction, again without training.

FreeFine. FreeFine [34] decouples geometric editing into three subtasks (transforming the object, inpainting the region it vacates, and refining the object at its target) and solves the last two with the same training-free diffusion procedure. Its authors attribute the difficulty of earlier methods under large transformations to the competing demands of solving all subtasks in one optimisation, for example preserving the background while generating newly exposed surfaces. They also introduce GeoBench (§4.7), on which FreeFine outperforms Self-guidance, RegionDrag, DragonDiffusion, Motion Guidance, DragDiffusion, Diffusion Handles, GeoDiffuser and DesignEdit. It is therefore the natural starting point for this thesis, and it is described in detail in Chapter 5.

Out-of-plane edits. Rotations out of the image plane reveal surfaces that the source never showed. They require a model of the object's three-dimensional shape, obtained from a novel-view synthesis model such as Zero-1-to-3 [35] or from lifting with estimated depth as in Diffusion Handles. GeoBench contains such edits in a separate three-dimensional part. This thesis studies the two-dimensional part, in which every edit is an in-plane affine transformation and the correct appearance of the object at its target pose is, up to resampling, available by warping the source.

4.4 Conditioning a frozen model

Several methods add new conditions to a frozen text-to-image model by training a small module once per kind of condition. ControlNet [36] trains a copy of the U-Net's encoder that receives a spatial map (edges, depth or a human pose) and adds its outputs to the skip connections of the frozen network through zero-initialised convolutions. T2I-Adapter [37] achieves spatial and colour control with lighter adapters that can be composed. IP-Adapter [38] adds image prompts through a decoupled cross-attention with about 22 million trained parameters, and DiffEditor uses a trained image encoder in the same role.

These modules are general-purpose: they supply information that the text prompt cannot, and they are reused unchanged across tasks. The thesis uses IP-Adapter and ControlNet as instruments rather than contributions. In Chapter 10 they supply the identity and the geometry of the object to FreeFine's refinement, which tests whether the refinement is missing either kind of information.

4.5 Guidance beyond classifier-free guidance

Classifier-free guidance (§3.7) improves the agreement with the prompt but, at high scales, over-saturates colours and introduces artefacts, and a line of work modifies the guidance update to avoid this. Imagen [39] introduced dynamic thresholding, which clips the predicted clean sample at each step. Kynkäänniemi et al. [40] show that guidance is harmful at high noise levels and unnecessary at low ones, and apply it only in an intermediate interval. CFG++ [41] attributes the failures of guidance to samples leaving the data manifold and reformulates guidance to stay on it, which also improves invertibility. Adaptive projected guidance (APG) [42] decomposes the guidance update into a component parallel to the conditional prediction and one orthogonal to it, identifies the parallel component as the source of over-saturation and down-weights it, and adds rescaling and a negative momentum across steps. Frequency-decoupled guidance (FDG) [43] splits the update into low and high spatial frequencies: the low frequencies govern global structure and agreement with the condition, the high frequencies visual detail, and the two receive separate scales.

These methods were developed to improve text-to-image generation over the whole image. In Free-Fine guidance is restricted to a region around the edited object (§5.4.4), and it plays a different role. Chapter 12 adapts APG and FDG to that setting and studies where the projection and the momentum of APG should be computed, which leads to the guidance rectifier used by the final method.

4.6 Feature-space and frequency-domain manipulation

The internal features of a diffusion U-Net are useful beyond generation. DIFT [25] shows that they establish accurate semantic and geometric correspondence between images without any training, which is why the keypoint metric of GeoBench uses them. Plug-and-Play, MasaCtrl and cross-image attention (§4.1) edit by injecting features or keys and values from one generation into another. FreeU [44] analyses the U-Net in frequency terms: the backbone contributes mostly to denoising, while the skip connections carry mostly high-frequency content, and re-weighting the two at inference improves generation quality without training.

This spectral view of the network's features motivates the mechanism of Chapter 13, which transfers only selected frequency bands of the reference features into the edited object instead of whole feature maps.

4.7 Evaluating geometric editing

The evaluation of editing combines three kinds of measurement. The realism of the edited images as a set is measured by distances between feature distributions: FID [45] on Inception features and the kernel distance [46], an unbiased estimate of the maximum mean discrepancy. Stein et al. [47] show that Inception-based metrics misjudge images produced by diffusion models and recommend DINOv2 features [48] instead. The consistency of the edit with its input is measured by the similarity of self-supervised or language-supervised features [10,49] on the object and on the background. The precision of a geometric edit is measured by where content ends up, as in the DIFT-based mean distance of DragBench (§4.2).

Table 4.1: Methods closest to this thesis. The last column marks the methods that FreeFine's authors compared on GeoBench.

Method	Training	How the edit geometry enters	On GeoBench
DragDiffusion [24]	per-image LoRA	latent optimisation towards dragged points	✓
DragonDiffusion [26]	none	gradient guidance from feature correspondence	✓
DiffEditor [27]	image encoder	gradient guidance with regional scores and image prompts	
RegionDrag [28]	none	copying latents from handle to target regions	✓
Self-guidance [29]	none	gradient guidance on attention-derived position and size	✓
Motion Guidance [30]	none	gradient guidance through an optical-flow network	✓
Diffusion Handles [32]	none	activations lifted to 3D with estimated depth	✓
GeoDiffuser [31]	none	transformation inside attention, with optimisation	✓
DesignEdit [33]	none	layered latents moved and fused	✓
FreeFine [34]	none	explicit warp, then inpainting and attention-controlled refinement	✓
This thesis	none	FreeFine with a behaviour selected by the requested transformation	✓

"Training" lists what must be trained or fine-tuned beyond the pretrained diffusion model; "none" means the method runs on the frozen model.

GeoBench [34] applies all three to geometric editing. It combines real and synthetic images from two existing collections into 811 source images and 5,988 editing instructions, of which 5,677 are two-dimensional: moves, rotations and resizings of one object, each with a known transformation and one of three intensity levels. Its seven metrics are defined in §5.7. Chapters 5 and 7 examine properties of those metrics that matter for interpreting any result on the benchmark.

4.8 Positioning of this thesis

Table 4.1 compares the methods closest to this thesis.

The thesis does not propose a new editing paradigm. It takes the strongest training-free method on GeoBench-2D, reproduces it, and asks why it falls short where it does. Four features distinguish it from the work above.

1. **It is led by a diagnosis.** Before any change, the failure of the reproduced method is located by transformation type and severity, and every later experiment is judged on the groups where the failure lies (Chapter 7).

2. **It tests explanations rather than accumulating mechanisms.** Six hypotheses about the cause of the failure are tested in turn, each with mechanisms drawn from the families of this chapter (configuration, conditioning, attention-level geometry, preservation of the coarse result,

guidance and feature manipulation), and the outcome of each experiment decides the next.

3. **Its final method is transform-conditional.** None of the methods in Table 4.1 changes its behaviour according to the kind of transformation it is asked to perform. The method of this thesis applies a different, experimentally supported behaviour to translations, rotations, moderate resizings and severe resizings, and it decides among them using only the requested transformation.

4. **It is evaluated at full scale against a reproduced baseline.** The final method is fixed on development subsets and then evaluated on all 5,677 two-dimensional edits with the official metric code, against the reproduced FreeFine run under identical conditions (Chapter 15).

Chapter 5

FreeFine and GeoBench-2D

This chapter describes the method that the thesis reproduces and extends, FreeFine [34], and the benchmark on which it is measured. It is the single place where the pipeline, its released configuration, the benchmark and the seven metrics are defined; every later chapter refers back to it. The presentation follows the published description, and it adds the details of the released implementation that the later experiments depend on.

5.1 Task and notation

Let $I_s \in [0,1]^{H\times W\times 3}$ be a source image and $M_s \in \{0,1\}^{H\times W}$ a mask selecting one object. A geometric edit is specified by a parameter vector

$$\theta = (d_x, d_y, d_z, r_x, r_y, r_z, s_x, s_y, s_z), \tag{5.1}$$

of which the two-dimensional benchmark uses the in-plane components: the translation (d_x, d_y) in pixels, the rotation r_z in degrees and the scale factors (s_x, s_y). The goal is an image $\hat{I}$ in which the object has undergone the requested transformation, the region it vacated has been filled plausibly, and the rest of the scene is unchanged. The method is *training-free*: every learned weight belongs to the frozen Stable Diffusion 1.5 model of §3.5, and the edit is obtained by manipulating how that model is sampled. Table 5.1 lists the symbols of this chapter; the diffusion notation is that of Table 3.2.

5.2 The decoupled pipeline

Earlier diffusion editors attempt the transformation of the object, the filling of the region it vacates and the harmonisation of the result within one optimisation or one guided sampling run, which becomes unreliable when the requested transformation is large (§4.3). FreeFine decouples the problem into three sequential stages (Figure 5.2).

1. **Object transformation.** The requested transformation is applied to the object, producing the object at its target pose and the target mask M_t. This stage is a deterministic geometric operation; no network is involved.

2. **Source-region inpainting.** The region vacated by the object is regenerated as background, giving I_{bg}.

Table 5.1: Notation for the editing pipeline, used throughout the thesis.

Symbol	Meaning
I_s, M_s, θ	source image, source object mask, edit parameters
A	2×3 affine matrix realising θ, Eq. (5.2)
M_t	target mask: the source mask warped by A
I_{bg}	background with the source region inpainted (stage 2)
I_c	coarse composite that enters refinement (stage 3), Eq. (5.3)
$\hat{I}$	edited image
i, t_i	index of a step on the 50-step grid and its timestep, $t_i = 981 - 20i$
s	index of the first refined step; the refinement runs $i = s, \ldots, 49$
γ_i	weight of the attention transplant at step i, Eq. (5.6)
m_t, m_s	target and source masks at the token resolution of an attention layer
m_{var}, m_{cfg}	regions of local perturbation and of guidance, at latent resolution

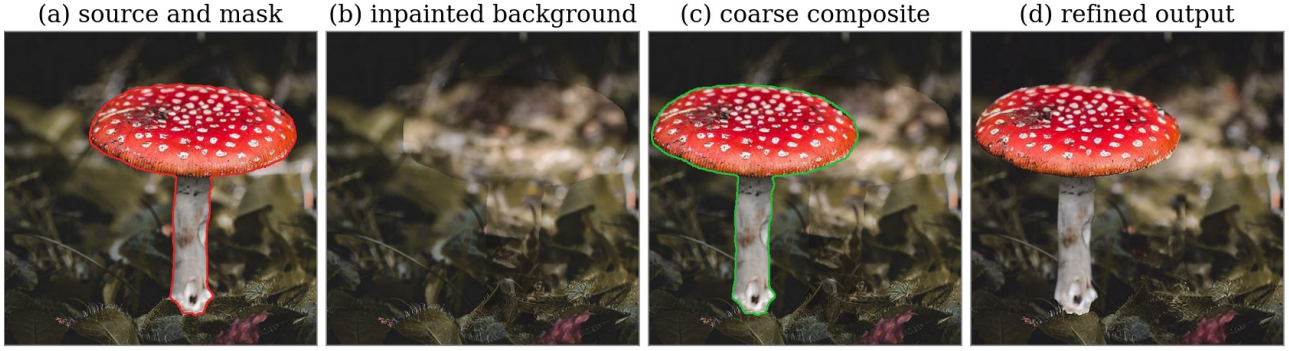


Figure 5.1: One edit of GeoBench-2D, a translation by 135 pixels to the left, through FreeFine. (a) The source image with the object mask. (b) The inpainted background, in which the region of the object has been regenerated. (c) The coarse composite: the transformed object placed on the inpainted background, with the target mask. (d) The output of refinement.

3. **Target-region refinement.** The transformed object is placed onto the inpainted background, and the resulting coarse composite I_c is refined: structures that the transformation could not supply are completed and the seams it introduced are removed.

Stages 2 and 3 are carried out by the same training-free diffusion procedure on the same frozen model. In the released benchmark code the inpainted background is produced once per object, with its own random seed, and stored; every edit of that object reads the same background, since all of them vacate the same region. Figure 5.1 follows one edit of GeoBench-2D through the three stages.

5.3 Object transformation and the coarse composite

For a two-dimensional edit the transformation is an affine map. With (c_x, c_y) the centre of the bounding box of the source mask, the implementation assembles

$$A = \begin{bmatrix} s_x \cos\rho & -\sin\rho & (1-\cos\rho)c_x + \sin\rho\, c_y + d_x + (1-s_x)c_x \\ \sin\rho & s_y \cos\rho & -\sin\rho\, c_x + (1-\cos\rho)c_y + d_y + (1-s_y)c_y \end{bmatrix}, \quad \rho = -\tfrac{\pi r_z}{180}, \tag{5.2}$$

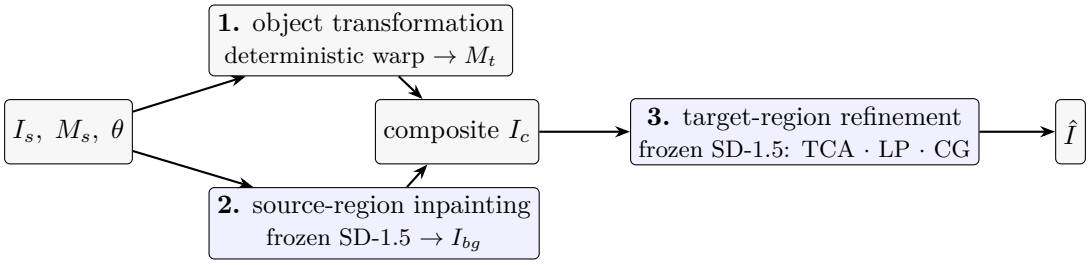


Figure 5.2: The decoupled pipeline of FreeFine [34]. Only the object transformation is deterministic; the inpainting of the source region and the refinement of the target region are both performed by the same training-free diffusion procedure on a frozen Stable Diffusion model.

that is, a rotation about the object centre, a scaling about the same centre, and a translation. Every benchmark edit activates exactly one kind of transformation, so the map reduces in each case to the intended operation: a pure translation $u \mapsto u + d$, a pure scaling $u \mapsto s(u - c) + c$, or a pure rotation about c. The image is resampled with bilinear interpolation and the mask with nearest-neighbour interpolation, and the coarse composite takes the warped object inside the target mask and the inpainted background outside it:

$$I_c = M_t \odot \mathcal{W}_A(I_s) + (1 - M_t) \odot I_{bg}. \tag{5.3}$$

The composite already shows the object at its exact target pose. It also carries the artefacts of the operation: bilinear resampling is not band-limited, so edges alias under reduction and rotation, and the insertion leaves a visible seam along the boundary of M_t.

5.4 Refinement

Refinement turns the coarse composite into a coherent image by partial denoising (§3.4), steered by three modules: Temporal Contextual Attention, Local Perturbation and Content-specified Generation.

5.4.1 Inversion and the four branches

The composite and the source image are encoded and inverted together with DDIM (§3.3) under the empty prompt, from the clean latent up to the entry step s of the 50-step grid. Refinement then runs the remaining steps $i = s, \ldots, 49$, at timesteps $t_i = 981 - 20i$. It maintains two latents:

- the **edit** latent, which starts from the inverted composite and becomes the output;
- the **reference** latent, which is replaced at every step by the inverted latent of the source image at the same timestep, so that it reproduces the source exactly and supplies its features to the attention layers.

Each step evaluates the network on a batch of four: an unconditional and a conditional prediction for the edit latent and for the reference latent, in the order unconditional edit, unconditional

reference, conditional edit, conditional reference. The conditional edit branch receives the object label as its prompt and the conditional reference branch the empty prompt. After the last step the edit latent is decoded, and the decoded image is the output $\hat{I}$; no part of the source is copied back.

5.4.2 Temporal Contextual Attention

Temporal Contextual Attention (TCA) transplants the appearance of the source object into the edit through mutual self-attention (§3.6). In each modulated self-attention layer, the queries $\mathbf{q}_p$ of the edit branch attend to the keys $\mathbf{k}_q$ and values $\mathbf{v}_q$ of the reference branch, in two masked forms:

$$S_o(p) = \sum_q \operatorname*{softmax}_q\Big(\tfrac{\langle \mathbf{q}_p, \mathbf{k}_q\rangle}{\sqrt{d}} + b^o_q\Big)\mathbf{v}_q, \qquad S_b(p) = \sum_q \operatorname*{softmax}_q\Big(\tfrac{\langle \mathbf{q}_p, \mathbf{k}_q\rangle}{\sqrt{d}} + b^b_q\Big)\mathbf{v}_q, \tag{5.4}$$

where $b^o_q = 0$ on the tokens of the source object and $-\infty$ elsewhere, and b^b_q is the opposite: S_o reads only from the source object and S_b only from the source background. With S_t the ordinary self-attention of the edit branch and m_t the target mask at the token resolution of the layer, the module outputs

$$H_i = (1-\gamma_i)\, S_t + \gamma_i\big[\, S_o \odot m_t + S_b \odot (1-m_t)\,\big]\,: \tag{5.5}$$

tokens inside the target region draw on the source object, tokens outside it on the source background, and both are blended with the edit's own self-attention. The weight γ_i decreases linearly over the refined steps,

$$\gamma_i = 1 - (1-e)\,\frac{i-s}{N-s}, \qquad i = s, \dots, 49, \tag{5.6}$$

with $N = 50$ and a floor parameter e. With the released $e = 0$ and $s = 35$, the weight falls from 1 at the first refined step to 1/15 at the last (Figure 5.3). Early steps therefore copy the structure of the source object into the target region, and later steps release the edit to its own self-attention, so that detail and boundaries can be harmonised with the scene.

TCA modulates the self-attention of the last six transformer blocks of the U-Net, blocks 10–15 of Table 3.1, which work at 32×32 and 64×64. The reference branches compute their attention without restriction, so that they keep reproducing the source image. Because the logits of Eq. (5.4) contain no positional term, TCA selects the source tokens that a query draws on by feature similarity alone (§3.6): the masks decide which part of the source a query may read from, not which token corresponds to it. The affine map of the edit, although known, does not enter the computation.

5.4.3 Local Perturbation

Refinement must be free to invent content where the edit requires it while leaving the rest of the image as it is. Local Perturbation makes the sampling step stochastic only in a region m_{var} at latent resolution, and deterministic elsewhere:

$$z_{t_{i+1}} = \sqrt{\bar{\alpha}_{t_{i+1}}}\,\hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t_{i+1}} - \sigma^2_{t_i}\, m_{\text{var}}}\;\hat{\varepsilon} + \sigma_{t_i}\,\xi \odot m_{\text{var}}, \qquad \xi \sim \mathcal{N}(0, \mathrm{I}), \tag{5.7}$$

where $\hat{\varepsilon}$ is the guided noise prediction and σ_{t_i} is the step noise of Eq. (3.10) with $\eta = 1$. Inside m_{var} the update is the stochastic step of §3.2; outside it, the deterministic DDIM step. The injected noise is the model's licence to synthesise new content, and it is granted only inside the region.

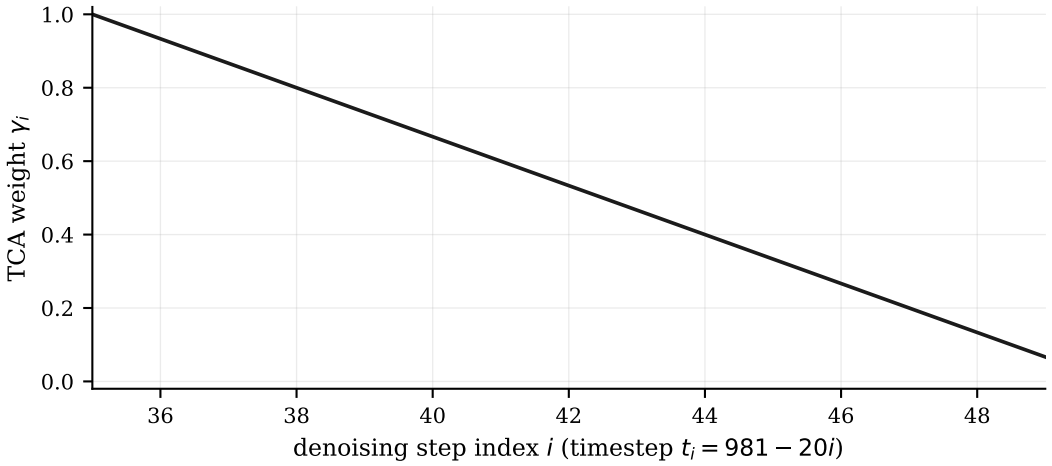


Figure 5.3: The weight γ_i of the attention transplant, Eq. (5.6), over the fifteen refined steps of the released configuration: the transplanted structure dominates at the start and has almost vanished at the last step.

Table 5.2: Settings of the released two-dimensional benchmark script of FreeFine (commit `4c9fdb97`).

Parameter	Value	Meaning
`num_step`	50	length of the DDIM grid, N
`start_step`	35	first refined step s; entry at $t_{35} = 281$, fifteen steps refined
`end_step`	50	end of the TCA schedule, Eq. (5.6)
`end_scale`	0.0	floor e of the TCA schedule
TCA blocks	10–15	modulated self-attention blocks, Table 3.1
`eta`	1.0	η of the stochastic step inside m_{var}, Eq. (5.7)
`guidance_scale`	7.5	w in Eq. (5.8)
`guidance_text`	`""`	object label p
regions	automatic	m_{var} and m_{cfg} from Eq. (5.9)
`seed`	42	random seed of the refinement stage
precision	fp32	single precision throughout

5.4.4 Content-specified Generation

Surfaces revealed by an edit have no counterpart in the source and must be generated. Content-specified Generation steers them with the object label p as a text prompt: its cross-attention is restricted to the target region, and classifier-free guidance (§3.7) is applied only in a region m_{cfg},

$$\hat{\varepsilon} = \varepsilon_{\varnothing} + w\,(\varepsilon_p - \varepsilon_{\varnothing}) \odot m_{\text{cfg}}, \tag{5.8}$$

so that newly generated content is consistent with what the object is.

5.5 The released configuration

Table 5.2 lists the settings of the released two-dimensional benchmark script, under which the published results are produced and which this thesis reproduces.

In this configuration the two regions of Local Perturbation and Content-specified Generation are not

Table 5.3: Composition of GeoBench-2D, computed from the exact edit parameters. Translations and rotations take integer values; scale factors are isotropic, $s_x = s_y$ in every case.

Edit type	Active parameters and observed range	Edits
Move	$d_x \in [-202, 194]$, $d_y \in [-203, 184]$ pixels	1,439
Rotate	$r_z \in [-39, 39]$ degrees	1,603
Resize	$s = s_x = s_y \in [0.400, 2.880]$	2,635
Total		5,677

drawn by a user but computed from the masks. Writing D_k for dilation by a $k \times k$ square, both are

$$m_{\text{var}} = m_{\text{cfg}} = \big(D_{30}(M_s) \cup D_{15}(M_t)\big) \setminus M_t, \tag{5.9}$$

taken at latent resolution: a band around the transformed object together with the area that the object vacated. The object itself lies outside both regions. Inside the target mask the refinement is therefore deterministic and unguided, and the object is shaped by the attention transplant; the stochastic, guided update acts on the boundary and on the vacated area, where new content has to be generated.

One entry of Table 5.2 decides what the released configuration computes. The label is the empty string, so the conditional prediction coincides with the unconditional one and the guidance term of Eq. (5.8) is zero whatever the scale (Eq. (9.2)). Content-specified Generation is therefore inactive in the released configuration. Experiment 5 verifies this on the outputs and switches the guidance on by supplying the object label.

5.6 GeoBench-2D

GeoBench [34] accompanies the method and covers two- and three-dimensional editing. It comprises 811 source images and 5,988 editing instructions, labelled by its authors as 2,267 easy, 2,075 medium and 1,646 hard, and divided into three subsets: 5,677 general two-dimensional edits, 190 three-dimensional edits, and 121 structure-completion instructions. Every instruction provides the object mask and the exact transformation parameters, so the target geometry is known for every case.

This thesis works with the two-dimensional subset, GeoBench-2D, which spans 609 source images and contains only in-plane transformations. Table 5.3 gives its composition, computed from the exact edit parameters of the released annotation.

Two features of the resizings matter later. Scaling is isotropic, so a single scalar $s = \sqrt{s_x s_y}$ describes the transformation exactly and is available from the requested parameters alone. And the distribution is bimodal: no edit has $s \in (0.9, 1.1)$, so the subset consists of genuine enlargements and reductions rather than near-identity perturbations.

5.7 The evaluation metrics

Seven metrics score an editing method along three dimensions [34]: the realism of the edited images as a set, the consistency of each edit with its source, and the precision of the edit. Table 5.4 summarises them.

Realism. FID compares Gaussians fitted to Inception-V3 features of the edited images and of a collection of real images [45,50]. The DINOv2 distance, written FID_{dino}, computes the same distance on features of a DINOv2 ViT-B/14 encoder [48], which judge images from diffusion models more faithfully than Inception features [47]. The kernel distance KD uses the same DINOv2 features but replaces the Fréchet distance by an unbiased polynomial-kernel estimate of the maximum mean discrepancy [46], averaged over one hundred random subsets.

Consistency. Subject consistency compares the edited object with the source object. Both are cut out with their masks, embedded with a self-supervised DINO ViT-B/16 encoder [49], and compared by cosine similarity:

$$\text{SUBC} = \frac{1}{N}\sum_{n=1}^{N} \max\Big(0,\ \cos\big(\phi_{\text{DINO}}(I_{s,n} \odot M_{s,n}),\ \phi_{\text{DINO}}(\hat{I}_n \odot M_{t,n})\big)\Big). \tag{5.10}$$

Background consistency (BGC) applies the same construction to the region outside both masks, with a CLIP ViT-B/32 image encoder [10]. Both are global descriptors of a masked image: they respond to changes of shape, colour and layout, and are comparatively insensitive to fine texture.

Precision. The warping error measures how far the edited object has moved away from the coarse composite inside the target region,

$$\text{WRAP_E} = \frac{1}{N}\sum_{n=1}^{N} \frac{\|(I_{c,n} - \hat{I}_n) \odot M_{t,n}\|_1}{\|M_{t,n}\|_1}, \tag{5.11}$$

and is minimised, by construction, when the output equals the coarse composite. Mean distance (MD), which this thesis also calls the keypoint error, measures whether content ended up where the transformation sends it. SIFT keypoints [51] of the source object that can be matched to the edited image (ratio test at 0.75, at most thirty per edit) are located in the edited image with DIFT correspondence [25] (features of Stable Diffusion 2.1 at timestep 261, averaged over eight noise samples and compared by cosine similarity), and the Euclidean distance between each located point $\hat{u}(k)$ and its analytic target $T(k)$ is averaged over all keypoints:

$$\text{MD} = \frac{1}{\sum_n |\mathcal{K}_n|}\sum_{n}\sum_{k\in\mathcal{K}_n} \|T(k) - \hat{u}(k)\|_2. \tag{5.12}$$

5.8 Properties of the metric implementation

Four properties of the released metric code determine what the seven numbers can and cannot express. They are part of the measuring instrument, and the later chapters account for each of them.

Table 5.4: The seven metrics as implemented. "Deterministic" means that re-evaluating the same images reproduces the value exactly.

Metric	Features	Compares	Better	Deterministic
FID	Inception-V3	edited set vs. real collection	↓	yes
FID_{dino}	DINOv2 ViT-B/14	edited set vs. real collection	↓	yes
KD	DINOv2 ViT-B/14	edited set vs. real collection, 100 subsets	↓	no
SUBC	DINO ViT-B/16	source object vs. edited object	↑	yes
BGC	CLIP ViT-B/32	source vs. edited background	↑	yes
WRAP_E	pixel ℓ_1	coarse composite vs. output, inside M_t	↓	yes
MD	SIFT + SD-2.1 features	located keypoints vs. analytic target	↓	no

The reference collection is fixed. For the three realism metrics the real collection is the entire content of a directory, not the set of sources paired with the edits being scored. When a subset of the edits is evaluated, the edited set shrinks while the reference set does not, so the value reflects the mismatch in size and content between the two collections as well as the quality of the edits. Values on different subsets are therefore not comparable with each other, only values of different methods on the same subset.

The keypoint target uses a different centre from the transformation. The analytic target $T(k)$ of Eq. (5.12) is obtained by rotating or scaling about the *centre of mass* of the object mask, whereas the transformation of Eq. (5.2) is applied about the *centre of the bounding box*. Writing δ for the difference between the two centres, the two conventions disagree by the constant vector $(\mathrm{I}-R_\rho)\delta$ under a rotation by ρ and $(1-s)\delta$ under a scaling by s. They agree for a translation, and for any object whose mask is symmetric about its bounding box. For rotations and scalings the metric therefore contains a case-dependent offset that is present even when the transformation is executed exactly.

The keypoint set depends on the image being scored. Keypoints are selected by matching the source against the *edited* image, so different methods are scored on different keypoints, in different numbers. Values of two methods are therefore not paired observations of the same quantity, and the value of a group is not a fixed-weight average of the values of its sub-groups, because the weights are the keypoint counts of each method.

Two of the seven metrics are stochastic. Subject consistency, background consistency, the warping error, FID and the DINOv2 distance are deterministic functions of the images. Mean distance is not, because its diffusion features are averaged over random noise samples, and the kernel distance is not, because it averages over random subsets.

Chapter 6

Methodology

This chapter sets out how the thesis works, before any experiment is reported. It states the problem and the questions that are asked (§6.1), the stages of the study (§6.2), the six hypotheses about the failure of FreeFine and the way each is tested (§6.3), the method to which the study leads (§6.4), the design of the evaluation (§6.5) and the implementation (§6.6). No measurement is reported here; every section closes with the places of the thesis where its subject is treated in full.

6.1 Problem statement and research questions

A geometric edit takes a source image I_s, a mask M_s that selects one object, and the parameters θ of a transformation: a translation, an in-plane rotation or an isotropic resizing (§5.1). The transformation maps the mask to a target mask M_t. The requirements on the output $\hat{I}$ (§2.1) fall into three groups, which have to be met at once.

- **Geometry.** The object stands where the transformation puts it: it fills M_t, and each of its points lies at the transformed position of the corresponding point of the source.

- **Identity.** The object is recognisably the object of the source image, with its shape, its texture and its detail.

- **Scene.** The region that the object vacated is filled plausibly, the rest of the image is unchanged, and the result is a realistic image without visible seams.

The three groups pull against each other. Copying the pixels of the object to the target pose keeps its identity and its geometry, but leaves seams and cannot supply what the transformation does not; regenerating the object blends it into the scene, at the risk of changing it. The seven metrics of GeoBench-2D measure the three groups separately: the warping error and mean distance measure the geometry, subject consistency the identity, and background consistency together with the three realism metrics the scene (§5.7).

The thesis studies this problem under fixed constraints. The generator is the frozen Stable Diffusion 1.5 (§3.5): no network is trained or fine-tuned, and everything that is added is computed while an edit is generated. The starting point is FreeFine as released (Chapter 5), the edits are the 5,677 in-plane edits of GeoBench-2D, and they are scored with the official metrics of the benchmark (§2.5).

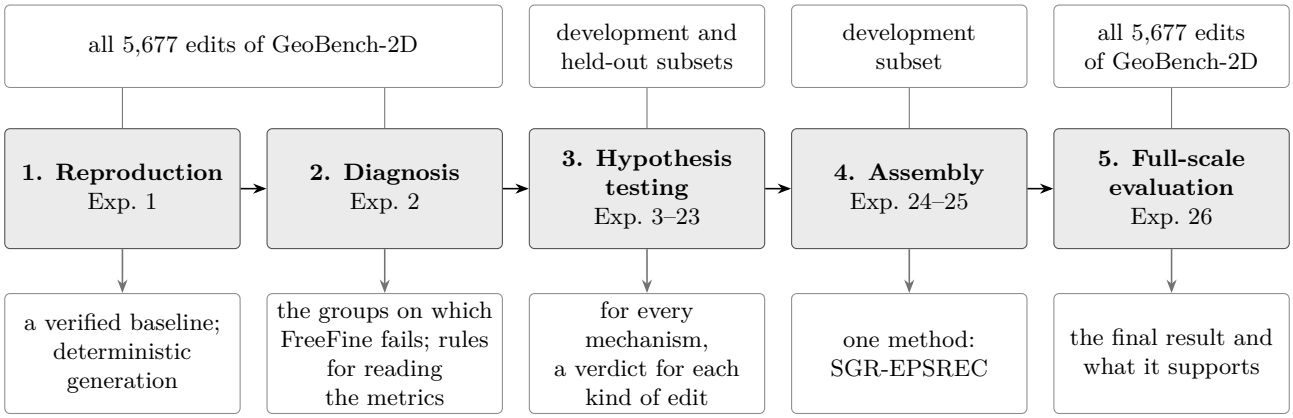


Figure 6.1: The five stages of the study. Top row: the edits on which each stage works. Bottom row: what each stage produces and hands to the next.

Within these constraints the thesis asks three questions (§2.3), and the rest of this chapter turns each of them into a procedure.

- **RQ1: can FreeFine be reproduced from its public code, and on which edits does it fail?** The released code is run on the whole benchmark and its results are dissected by kind of edit (Stages 1 and 2 of §6.2).

- **RQ2: what causes the failure?** Six possible causes are stated as hypotheses, and each is tested with controlled changes to the pipeline (Stage 3, §6.3).

- **RQ3: can FreeFine be improved without training, and how much of the improvement reflects better editing?** The components that the tests support are assembled into one method (Stage 4, §6.4), which is evaluated on the whole benchmark together with an analysis of what each metric can show (Stage 5, §6.5).

Where to find the details. The task and its notation: §5.1 and Table 5.1. The three stages of FreeFine: §5.2. The metrics: §5.7. The scope of the thesis: §2.5. Earlier work on the problem: Chapter 4.

6.2 Overview of the study

The study proceeds in five stages, each of which produces what the next one needs (Figure 6.1, Table 6.1).

1. **Reproduction (Experiment 1).** The released code of FreeFine is run on all 5,677 edits without any change to what it computes, and the seven metrics are compared with the published values. The stage also checks that generation is deterministic: the same inputs give the same image, pixel for pixel. It produces a baseline whose values are known, and the guarantee on which every later comparison rests: two results that differ in one setting can be compared image by image, and their difference is caused by that setting.

2. **Diagnosis (Experiment 2).** The reproduced results are scored separately by the type of the edit and by its severity, for which a severity score is defined from the requested edit alone (Eq. (7.2)).

Table 6.1: The stages of the study: the question each answers, the edits it uses, its experiments and the place where it is reported.

Stage	Question	Edits	Exp.	Reported in
1. Reproduction	Does the released code return the published results? (RQ1)	all 5,677	1	§7.1
2. Diagnosis	On which edits, and on which metrics, does FreeFine fail? (RQ1)	all 5,677	2	§7.2
3. Hypothesis testing	What causes the failure? (RQ2)	development (200), held-out (200)	3–23	Chapters 9–13
4. Assembly	Can the supported components be combined into one method? (RQ3)	development (200)	24–25	Chapter 14
5. Full-scale evaluation	Does the method improve FreeFine, and what do its gains mean? (RQ3)	all 5,677	26	Chapters 15, 16

The released metric code is examined at the same time, to establish what each of the seven numbers can and cannot express. The stage produces the target of the study, the groups of edits on which FreeFine fails (the hard rotations and the hard resizings), and the rules by which the metrics are read.

3. **Hypothesis testing (Experiments 3 to 23).** Six explanations of the failure are stated as hypotheses, and each is tested by changing the pipeline at the place that the hypothesis names (§6.3). Every test runs on a development subset of 200 edits under one protocol (§6.5), and the first combined models are checked on a second, disjoint subset. The stage produces, for every mechanism, a verdict for each kind of edit: where it helps, where it harms and where it changes nothing.

4. **Assembly (Experiments 24 and 25).** The mechanisms that the tests support are combined into one method. Because each of them suits only some edits, the method chooses among them, and the choice is made by a rule that reads nothing but the requested transformation (Experiment 24). The assembled method is then run on the development subset, to check that each of its branches behaves as it did in the experiment that established it (Experiment 25).

5. **Full-scale evaluation (Experiment 26).** The method and FreeFine are generated in the same run on all 5,677 edits and scored on the whole benchmark and on each branch of the method. The values of the two models are compared edit by edit, and the gains are examined one by one, to separate those that reflect better editing from those that follow from how the method or a metric is built.

The order of the stages is also the order of the claims: nothing is proposed before the baseline is verified, no mechanism is tested before the failure is located, and the method is assembled only from components that were tested in isolation.

Where to find the details. Reproduction and diagnosis: Chapter 7. The protocol of the experiments: Chapter 8. The six hypotheses: Chapters 9 to 13. The assembly: Chapter 14. The full-scale evaluation: Chapter 15. The synthesis of all results: Chapter 16. The chapters at a glance: Figure 2.1.

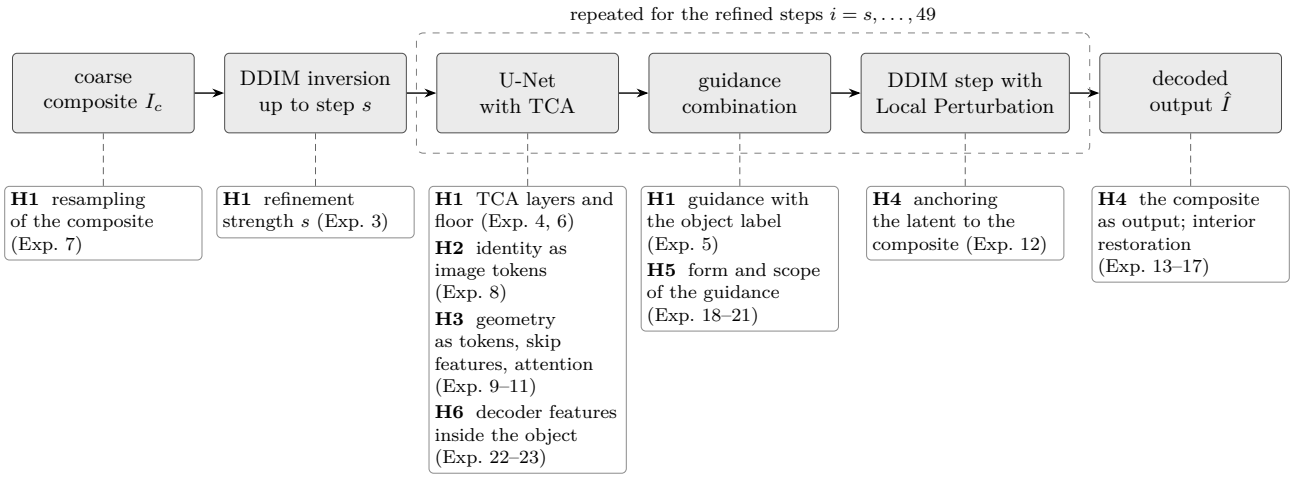


Figure 6.2: The refinement of FreeFine as a sequence of operations (§5.4), and the place at which each hypothesis changes it. The dashed frame encloses the operations that are repeated at every refined step.

6.3 The six hypotheses and how each is tested

FreeFine's refinement receives three things: the coarse composite, in which the object already stands at its target pose; the source image, whose features the attention transplant copies; and an empty prompt (§5.4). It turns them into the output by a fixed sequence of operations (Figure 6.2). A failure on the hard groups can therefore originate in a small number of places, and each hypothesis names one of them.

- The *settings* with which the sequence is run may be badly chosen (Hypothesis 1).
- The refinement may *lack information* that it is never given: what the object is (Hypothesis 2), or how it was transformed (Hypothesis 3).
- The refinement may *lose information* that its input already contains, since the composite shows the object at its exact target pose (Hypothesis 4).
- Where the transformation is severe and the object has to be reconstructed at a new scale, the *reconstruction may be badly steered*: around the object, where guidance acts (Hypothesis 5), or inside it, where no guidance reaches (Hypothesis 6).

The hypotheses are ordered from the least to the most invasive change, and each is examined before the next mechanism is added: a setting is changed before anything is added to the pipeline, information is supplied before the treatment of the composite is changed, and the general treatment is settled before the special case of severe resizings is addressed. Table 6.2 lists what is changed for each hypothesis, and Figure 6.2 shows where each change acts.

Hypothesis 1: tuning the released pipeline. Five experiments vary, one at a time, the settings that the released pipeline exposes and the quality of its input: the refinement strength (Experiment 3, §9.1), the attention layers that TCA modulates (Experiment 4, §9.2), guidance with the object label (Experiment 5, §9.3), the floor of the TCA schedule (Experiment 6, §9.4) and the re-

Table 6.2: The six hypotheses, what is changed to test each, where the change acts, and where it is reported.

	Claim	What is changed	Acts on	Exp.	Ch.
H1	A better setting of the released pipeline exists	refinement strength, TCA layer range, guidance with the object label, floor of the TCA schedule, resampling of the composite	settings of the pipeline	3–7	9
H2	The refinement lacks the identity of the object	a crop of the source object, given as image tokens (IP-Adapter) at four scales	cross-attention	8	10
H3	The refinement lacks the geometry of the edit	the transformed object, or an edge drawing of it, as image tokens; the edges of the composite through a ControlNet; four changes that make the attention transplant aware of the transformation	cross-attention, skip connections, self-attention	9–11	10
H4	The refinement loses what the composite gets right	anchoring of the latent to the composite's trajectory; the composite scored as an output; interior restoration after refinement (width of the band, scope, kind of edit); first routed models	latent in the band around the object; output image	12–17	11
H5	Severe resizings need a better-steered reconstruction	projected guidance on the noise predictions and on the predictions of the clean image, guidance split by spatial frequency, scope of the guidance	guidance combination, in the band around the object	18–21	12
H6	The structure of the composite must be given inside the object	spectral fusion of decoder features: frequency band, reference, with and without guidance	decoder features inside the object	22–23	13

sampling that produces the composite (Experiment 7, §9.5). The hypothesis is supported if some setting raises subject consistency on a hard group clearly above the baseline without a loss elsewhere. Whatever the verdict, the experiments also measure how far the metrics move when nothing new is added to the system, and this spread fixes the threshold that every later mechanism has to clear (§9.6).

Hypotheses 2 and 3: supplying information the model lacks. A frozen U-Net can receive information from outside at three places: the tokens read by cross-attention, the feature maps added to the skip connections, and the keys and values of self-attention (Figure 10.1). The identity of the object is supplied as image tokens computed from a crop of the source object (Experiment 8, §10.1). The geometry of the edit is supplied through all three places: as image tokens of the transformed object (Experiment 9, §10.2), as the edges of the composite through a ControlNet (Experiment 10, §10.3), and inside the attention transplant itself, in four ways that make it aware of the transformation (Experiment 11, §10.4). Each hypothesis is supported if its mechanism raises subject consistency on a hard group by more than the threshold or, for the geometry, clearly lowers mean distance, without a loss on translations or on the background.

Hypothesis 4: anchoring the refinement to the composite. The composite is scored as if it were an output (Experiment 13, §11.2), which measures how much it contains that refinement could lose. Two ways of keeping the output closer to it are tested: during denoising, by pulling the latent towards the composite's own trajectory after every step (Experiment 12, §11.1); and after refinement, by restoring the interior of the object from the composite while a band along its outline keeps the refined result, in first models that choose the treatment by the type of the edit (interior restoration, Experiment 14, §11.3). Three further experiments establish how wide the band should be (Experiment 15, §11.4), whether the gain comes from where restoration is applied or from the quality of the composite (Experiment 16, §11.5), and for which kinds of edit restoration helps (Experiment 17, §11.6). The hypothesis is supported if keeping the output closer to the composite improves the edit without lowering realism.

Hypothesis 5: guiding the reconstruction of severe resizings. A severe resizing asks refinement to reconstruct the object at a scale at which the source never showed it. The experiments change the form of the guidance that steers this reconstruction in the band around the object: adaptive projected guidance on the noise predictions (Experiment 18, §12.1), guidance split into low and high spatial frequencies (Experiment 19, §12.2), projected guidance on the predictions of the clean image, with controlled variants that isolate each of its operations and lead to the EPSREC formulation (Experiment 20, §12.3), and the set of edits to which guidance is applied (Experiment 21, §12.4). They are judged by mean distance on the hard resizings, where a change counts only if it exceeds the variation that the metric shows between near-identical variants (Experiment 3).

Hypothesis 6: supplying the composite's structure inside the object. Guidance does not act inside the object (Eq. (5.9)). Spectral feature fusion mixes selected frequency bands of the U-Net's decoder features, inside the object, with the features that the same network computes from the composite (Experiment 22, §13.1). A second experiment varies the band and the reference, and runs every variant with and without guidance and in combination with EPSREC (Experiment 23, §13.2). The criterion is that of Hypothesis 5.

The form of every experiment. The twenty-six experiments of the thesis are numbered once and reported in the same three parts: what is changed and why, with the theory needed to understand it; the measurements on the standard groups; and what they establish. Each experiment changes one thing with respect to a stated reference model. A mechanism that does not help is reported as completely as one that does, because a rejected explanation narrows the search as much as an accepted one.

Where to find the details. The three entry points of a frozen U-Net: Figure 10.1. The two places where the reconstruction is steered: Figure 12.1. The verdict on each hypothesis and the evidence for it: §16.1 and Table 16.1.

6.4 The method in overview

The study leads to one method, SGR-EPSREC: Selective Geometric Refinement with EPSREC guidance. Its components were not fixed in advance; each is kept because an experiment of Stage 3 estab-

lished it for one kind of edit. The method is summarised here so that the reader knows where the experiments lead. Its definition is Eq. (14.2), and Figure 14.1 shows it as a diagram.

The principle. FreeFine refines every coarse composite in the same way. SGR-EPSREC treats the composite according to the transformation that produced it: it keeps the interior of the composite where the composite can be trusted, leaves FreeFine's refinement untouched where refinement repairs the composite, and steers the reconstruction where the transformation is severe.

Interior restoration. For translations and moderate resizings the output of FreeFine is post-processed. The interior of the object, obtained by eroding the target mask by w pixels, is taken from the coarse composite, and the band of width w along the outline keeps the refined result (Eq. (11.2), Figure 11.2). The interior then carries the source object as the transformation placed it, and the band carries the blending with the scene that only refinement provides. The operation needs no generation of its own. The width is $w = 4$ pixels for translations and $w = 8$ for moderate resizings.

EPSREC. For severe resizings refinement is run with the object label as its prompt, and the rule that combines the unguided and the label-conditioned predictions in the band around the object is replaced. EPSREC accumulates the label's correction over the refined steps in a recursive memory with a negative momentum, kept in the space of the noise predictions; it converts the memory to the space of the predicted clean image, removes the part that is parallel to the conditional prediction, and applies the result with a guidance scale of $w = 10$, in the band only (§12.3). The negative momentum damps the push that guidance repeats at every step and keeps the part of the correction that changes from step to step. EPSREC changes only how the two predictions that FreeFine already computes are combined: the attention transplant, the regions and the sampler are untouched.

The routing rule. The branch of an edit is chosen from its requested parameters alone. The type of the edit is read from θ, and a resizing with scale $s = \sqrt{s_x s_y}$ is severe if $s \geq 1.5$ or $s \leq 0.6$, the range that the authors of GeoBench define as a hard resizing (Eq. (14.1)); no metric of any method was consulted to set the thresholds. Translations and moderate resizings receive interior restoration, severe resizings are generated with EPSREC, and rotations, which no component of the study improves, are left exactly as FreeFine produces them (Table 14.1).

Properties. The method trains no parameter and uses no learned router. It reads only what a user of FreeFine already provides: the image, the mask and the transformation. With every switch off it is FreeFine, bit for bit. One optional addition, spectral fusion in the branch of the severe resizings, is carried as an ablation and evaluated beside the method (Experiment 25 and Chapter 15).

Where to find the details. Interior restoration: §11.3 (definition), §11.4 (width of the band), §11.5 (what makes it work), §11.6 (kinds of edit). EPSREC: §12.3 (derivation and the eight steps of the algorithm), §12.4 (where it is applied); background on guidance: §3.7 and §4.5. The routing rule: §14.2. The method and its check on the development subset: §14.3 and §14.4. Example edits of every branch: Figures 14.2 and 14.3.

Table 6.3: The three sets of edits on which results are measured.

Edits	Size	Used for	Defined in
GeoBench-2D	5,677	reproduction, diagnosis and final evaluation (Experiments 1, 2 and 26)	§5.6
Development subset	200	every hypothesis and the assembly (Experiments 3 to 25)	§8.1
Held-out subset	200	check of the composite and of the first combined models (Experiments 13 and 14)	§8.1

6.5 Evaluation design

Benchmark and metrics. All results are measured on GeoBench-2D: 5,677 edits of 609 images (1,439 translations, 1,603 rotations and 2,635 resizings), each with its object mask and its exact parameters (§5.6). An output is scored by the seven metrics of the benchmark, computed with the authors' code. FID, the DINOv2 distance and the kernel distance measure the realism of the edited images as a set. Subject consistency and background consistency measure how well the object and the scene are kept. The warping error and mean distance measure the precision of the edit: the first as the distance of the output from the coarse composite inside the target mask, the second as the distance of the object's keypoints from the positions to which the transformation sends them (Table 5.4).

Three scales. The edits on which a result is measured depend on its purpose (Table 6.3). The whole benchmark is used where the published values are defined and where the realism metrics can be read: for the reproduction, the diagnosis and the final evaluation. Testing a hypothesis, however, means regenerating every edit for every variant, so the hypotheses are tested on a *development subset* of 200 edits, drawn with a fixed seed so that the nine groups of edit type and severity are equally represented. A *held-out subset* of 200 other edits, never used to develop a method, checks that a result obtained on the development subset is not tied to its edits.

Groups. Every variant is reported on four groups of the development subset: the hard rotations and the hard resizings, which are the target; all translations, which FreeFine already handles well and which serve as a control against harm; and the whole subset, on which background consistency and realism are computed (Table 8.2). From the assembly onward the groups are the branches of the method itself: translations, rotations, moderate resizings and severe resizings.

Controls. Three controls accompany every study. Each modification sits behind a switch, and before a study starts the modified code, with every switch off, must reproduce the stored baseline images exactly (the *switch-off check*). Every variant is generated from the same inpainted backgrounds and with the same seed as the baseline, so that two variants are *paired* image by image. And the images of every variant are compared with those of the baseline, to confirm that the modification has acted at all (*liveness*). The two metrics that draw random numbers, mean distance and the kernel distance, are evaluated with a fixed seed, so that the same images always receive the same value.

When a change counts. Every result is stated together with the group it holds on and the size of that group. The stochastic metrics are compared only within one seeded evaluation, and the realism metrics only between models scored on the same edits. A metric that a method improves by construction, for instance the warping error of any method that copies the composite into its output, is reported but not used as evidence. The threshold for an improvement of subject consistency is not chosen in advance; it is derived from Hypothesis 1. A mechanism must raise subject consistency on a hard group by more than 0.005, about twice the largest change that re-configuring the released pipeline produces, without a loss on translations or on the background (§9.6). Mean distance is judged against the variation that it shows between near-identical variants (Experiment 3).

The full-scale evaluation. In the final evaluation FreeFine is generated again together with the method, and its new images are checked to be identical to those of the reproduction. All seven metrics are computed on the whole benchmark and on each branch. Because the two models produce every edit from the same inputs, subject consistency is also compared edit by edit: the mean difference is reported with a bootstrap confidence interval, together with the share of edits that improve and that get worse, and with a signed-rank test where the difference is small. Each gain is then examined for what it means: whether the method guarantees it by construction, whether it depends on how a metric weights the edits, and whether it survives the removal of the development edits.

Where to find the details. The benchmark and its composition: §5.6 and Table 5.3. The definitions of the metrics: §5.7; the properties of their implementation: §5.8. The complete protocol: Chapter 8 (subsets §8.1, groups §8.2, controls §8.3, seeding §8.4, reporting rules §8.5). The severity score: §7.2.1. The setup and the analysis of the final evaluation: §15.1 and §15.3.

6.6 Implementation and reproducibility

The thesis is built on the public code of FreeFine. Five changes were needed to run it outside its authors' environment, none of which alters what the model computes; generation and evaluation run in two separate software environments, on NVIDIA T4 GPUs and in single precision (§7.1). Every mechanism of the thesis is added to this code behind a switch of its own, so that one code base produces the baseline and every variant, and the switch-off check of §6.5 ties each study to the same baseline. Generation is deterministic: the random seed of the refinement is fixed, and the inpainted backgrounds are computed once and stored. All scoring uses the authors' evaluation code with the two changes of Experiment 1. The code, the executed notebooks and the results of every experiment are public.[1]

Where to find the details. The environments, the changes to the released code and the check of determinism: §7.1 and Tables 7.1 and 7.2. The released settings of FreeFine: Table 5.2. The provenance of the code and of the scoring: §8.6.

[1] https://github.com/giorgostzamouranis/Diploma-Thesis

Chapter 7

Reproduction and Diagnosis

Before FreeFine can be improved, two questions must be answered. Can the published system be run outside its authors' environment, and does it then return the published numbers? And where are its errors concentrated: in which kind of edit, at which severity, on which metric? The first question is a precondition for everything that follows, because a change to the method can only be credited with an effect if the value it changes is known. The second decides which explanations are worth testing. Experiment 1 reproduces FreeFine on all 5,677 edits of GeoBench-2D; Experiment 2 dissects the reproduced results by edit type and severity.

7.1 Experiment 1: Reproducing FreeFine

7.1.1 What was done

Environment. All generation and evaluation ran on NVIDIA T4 GPUs with 15 GB of memory. Generation and evaluation cannot share a software environment: the released inference code targets version 0.18.0 of `diffusers`, whose interfaces changed in later releases, while the evaluation code needs `torch` 2.6 and CLIP and dataset libraries that conflict with the generation pins. Two environments were therefore built on the same Python version (Table 7.1). Both diffusion checkpoints are referenced at locations that no longer resolve (SD-1.5 at an absolute path on the authors' machine, and the SD-2.1 model used by the keypoint metric at a withdrawn repository), so they were loaded from their current official mirrors, which redistribute the same weights.

Changes needed to run the released code. The released two-dimensional scripts do not run unmodified. Table 7.2 lists the five changes that were necessary. Two of them repair defects. In the inference script the coarse composite is built by a call that reads the source mask one line before that mask is assigned, so the first edit raises an error and every later edit would be composed with the previous edit's mask; the two lines were reordered. In the evaluation entry point an argument is read as `args.3d`, which is not valid Python, so the module fails before any metric is computed; it was replaced by `getattr(args, "3d")`. None of the five changes alters what the model computes.

One further change touches the computation and therefore needs an argument. Masked attention is evaluated over a tensor of shape $(B \cdot H) \times N_q \times N_k$, which exceeds the memory of the GPU at 64×64. The implementation was modified to process the batch-times-heads axis in chunks of eight. The softmax of attention is taken along the key axis for each query row separately, and chunking

Table 7.1: The two software environments. The generation column follows the authors' requirements; the only addition is the `setuptools` ceiling, which restores a module that `pytorch-lightning` 2.1.3 imports.

	Generation	Evaluation
Python	3.10.13	3.10.13
`torch`	2.1.1+cu121	2.6.0+cu124
`diffusers`	0.18.0	not used
`xformers`	0.0.23	not used
`transformers`	4.30.1	as required by the metric code
Additional pins	`pytorch-lightning` 2.1.3, `setuptools<70`	CLIP at commit `dcba3cb`, `pyarrow<16`, `datasets<3`
Diffusion weights	SD-1.5, single precision	SD-2.1, features for MD

Table 7.2: The five changes required to run the released two-dimensional pipeline, and the memory change discussed in the text.

Change	Component	Reason
Source mask loaded before it is used	inference script	the mask is read one line before it is assigned
`args.3d` replaced by `getattr`	metric entry point	not a valid Python attribute reference
Two keyword arguments removed	background-generation script	not accepted by the method they are passed to
Weight locations rewritten	inference script and MD feature extractor	the referenced locations no longer resolve
Dataset paths and one file name	all four batch scripts	absolute paths on the authors' machine; the annotation file name differs by one character
Masked attention computed in chunks[a]	attention module	peak memory exceeds 15 GB

[a] Leaves every value unchanged; see the text.

partitions the rows without reordering any reduction, so the result is numerically identical and only the peak memory changes.

The inpainted backgrounds. The released benchmark code does not run the pipeline of §5.2 end to end for each edit. One script inpaints the source region of each object and writes the result to disk, and the editing script reads it back. The stored result depends only on the image and the object, so all edits of the same object share one background, which turns the 5,677 edits into 668 background generations. The background script draws a fresh random seed for every object, while the editing script fixes its seed at 42, and the benchmark distributes the source images, masks and coarse composites but not the backgrounds. The 668 backgrounds were therefore generated once and then held fixed: every generation run in this thesis uses the same set, exactly as the released code intends (Figure 7.1).

Because the set is identical for every method compared in this thesis and generation downstream of it is deterministic, the particular draw cancels from every comparison made here. It does not cancel from the comparison with the published results, since the authors' draw is not available.

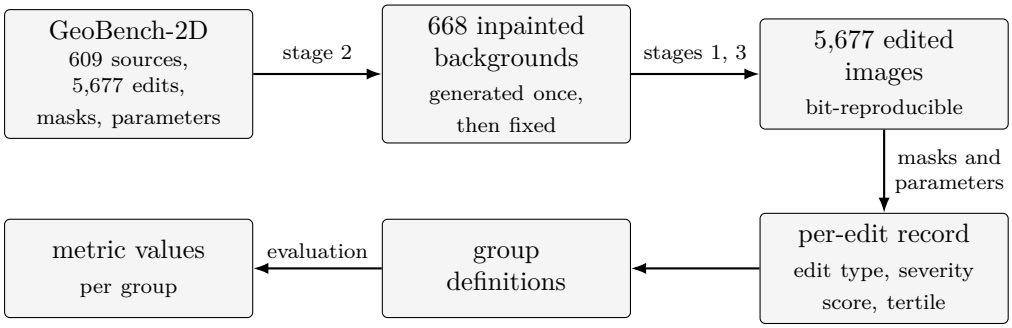


Figure 7.1: The artefacts of the reproduction. The inpainted backgrounds are generated once, with the random seeds of the released code, and every later generation run reuses them; everything downstream is deterministic.

Table 7.3: The reproduced baseline against the published two-dimensional results of FreeFine [34], over all 5,677 edits. Deviations are relative to the published value.

Metric	Published	Reproduced	Δ	Δ (%)
FID ↓	34.72	35.0439	+0.32	+0.9
FID_{dino} ↓	478.18	487.8231	+9.64	+2.0
KD ↓	0.144	0.1423	−0.0017	−1.2
SUBC ↑	0.907	0.9113	+0.0043	+0.5
BGC ↑	0.971	0.9670	−0.0040	−0.4
WRAP_E ↓	0.055	0.0474	−0.0076	−13.9
MD ↓	9.25	8.4993	−0.75	−8.1

Mean distance is the exact keypoint-pooled value of the first evaluation; Table 7.4 gives four evaluations of the same images.

Determinism. With the backgrounds fixed, generation is deterministic: regenerating stored edits with the same code reproduces the stored images exactly, with a mean absolute pixel difference of 0.0. Two results generated with the same inputs and differing in one setting are therefore paired image by image, and a difference between them is caused by that setting alone.

Evaluation. All scoring uses the authors' evaluation script with its default reference directory for the realism metrics and with the difficulty filter disabled, so that no edit is excluded. Of the nine metrics that the script implements, the seven of Table 5.4 were computed; the other two, an instruction-following score and a human-preference score, are not part of the published two-dimensional comparison.

7.1.2 Results

Table 7.3 places the reproduction beside the published two-dimensional results, and Figure 7.2 shows the relative deviations.

Five of the seven metrics agree with the published values to within two per cent. The two exceptions are the metrics of geometric precision, and both lie on the favourable side: the warping error is 13.9% and the keypoint error 8.1% below the published values. The largest unfavourable deviation is +2.0%,

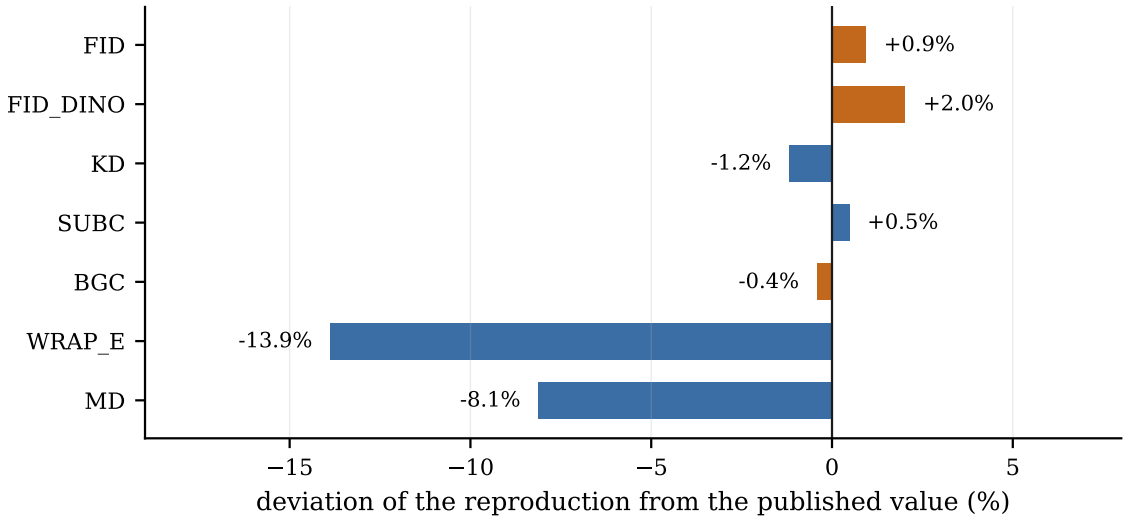


Figure 7.2: Deviation of the reproduction from the published value, per metric. Darker bars mark metrics on which the reproduction is on the better side of the published value.

Table 7.4: Four evaluations of mean distance on the same 5,677 reproduced images.

Evaluation	Aggregation	MD
First run, two parts	part means combined by edit count	8.4636
First run, keypoint dump	exact pooling over keypoints	8.4993
Second run, keypoint dump	exact pooling over keypoints	8.5455
Final run, seeded	as computed by the script	8.5024

Both keypoint dumps contain exactly 120,155 keypoints over 5,673 edits; in four edits no keypoint survives the ratio test.

on the DINOv2 distance.

The keypoint metric on repeated evaluations. The same 5,677 images were scored for mean distance four times (Table 7.4). The images are bit-identical in all four rows; only the evaluation differs.

7.1.3 Outcome

The reproduction is faithful. The reproduced system is close to the published one on realism and consistency and better on geometric precision, so the rest of the thesis studies FreeFine itself and not a degraded copy of it. The favourable deviations are not attributed to a cause. The weights, the inference script and all its settings, the benchmark and the evaluation code were held fixed; the inpainted backgrounds, the GPU and driver, and the exact binaries of the libraries could not be, since the authors' versions of them are not available. Separating these factors would require controlled full-scale runs that vary one at a time, and none was carried out. For the same reason the reproduced results, not the published ones, serve as the baseline: every comparison in this thesis is between results produced by the same code, on the same hardware, from the same backgrounds.

Mean distance is a stochastic measurement. Table 7.4 shows two separate effects. The first two rows come from the same execution and differ by 0.036 only because of how the result was aggregated: the metric is a mean over keypoints, and combining the means of two parts by their number of edits assumes that every edit contributes the same number of keypoints. The second effect is genuine noise. The two keypoint dumps contain exactly the same 120,155 keypoints, so keypoint detection is deterministic, but their localisation uses diffusion features averaged over random noise, and re-scoring identical images moved the pooled value by 0.046, about half a per cent. Two consequences are carried forward. A difference of the order of 0.05 on the full set carries no information unless both values come from the same seeded evaluation; the random state of the metric is therefore fixed from the next chapters on (Chapter 8). And because every method is scored on its own keypoints, the value of a group cannot be predicted from the values of its parts.

With a faithful baseline in hand, the next question is where it fails.

7.2 Experiment 2: Where FreeFine fails

7.2.1 What was done

The reproduced results were scored separately by edit type and by severity. The benchmark carries its own difficulty label, but a severity measure was defined here instead, for two reasons: it can be computed for any subset from the requested edit alone, including the subsets built in later chapters, and its definition is fixed by this work. The per-severity numbers of this thesis are therefore not comparable with per-difficulty numbers reported elsewhere, including the benchmark's own.

The severity score. For an edit with parameters θ (Eq. (5.1)), source mask M_s and image size $H \times W$, four quantities are computed,

$$t = \frac{\|(d_x, d_y)\|_2}{\sqrt{H^2 + W^2}}, \qquad r = |r_z|, \qquad \sigma = |\ln s_x| + |\ln s_y|, \qquad a = \frac{|M_s|}{HW} : \tag{7.1}$$

the translation as a fraction of the image diagonal, the rotation angle, the logarithmic scale change and the fraction of the image covered by the object. Each is min-max normalised over the 5,677 edits, written $\hat{\cdot}$, and the severity score is their mean,

$$D = \tfrac{1}{4}(\hat{t} + \hat{r} + \hat{\sigma} + \hat{a}). \tag{7.2}$$

The edits are split at the 1/3 and 2/3 quantiles of D into 1,892 easy, 1,892 medium and 1,893 hard edits.

Every edit of GeoBench-2D activates exactly one kind of transformation, so only the term of that kind and the object-size term are non-zero. The score is therefore not a like-for-like severity across edit types, and the tertiles are not balanced across them (Table 7.5): 42.4% of the rotations fall in the hard tertile, against 28.7% of the translations. For this reason every result by severity is reported together with the edit type, and the score is used to locate failures, not to measure intrinsic difficulty.

Table 7.5: The severity tertiles of Eq. (7.2) against the edit type, over all 5,677 edits.

	easy	medium	hard	total
Move	484	542	413	1,439
Rotate	434	489	680	1,603
Resize	974	861	800	2,635
Total	1,892	1,892	1,893	5,677

Table 7.6: The reproduced baseline on the three edit types and on all edits.

Group	n	SUBC↑	BGC↑	WRAP_E↓	MD↓	FID↓	FID_{dino}↓
Move	1,439	0.9592	0.9670	0.0489	3.84	47.40	580.83
Rotate	1,603	0.9008	0.9673	0.0433	10.47	47.55	588.75
Resize	2,635	0.8916	0.9669	0.0490	9.99	40.85	531.29
All	5,677	0.9113	0.9670	0.0474	8.55	35.04	487.82

Mean distance comes from the second evaluation of Table 7.4, the one whose per-group values were recorded.
The realism columns are not comparable across rows (§5.8).

7.2.2 Results

By edit type. Table 7.6 and the left half of Figure 7.3 give the results for the three edit types.

By severity. Table 7.7 gives the results for the three severity tertiles. Figure 7.4 crosses severity with edit type on the keypoint metric, and Figure 7.5 shows the realism metrics across all groups.

7.2.3 Outcome

Translation is close to solved; rotation and resizing are not. Subject consistency is 0.9592 on translations and 0.9008 and 0.8916 on rotations and resizings, a gap of 0.058 and 0.068. On the keypoint metric the gap is larger still: 3.84 pixels against 10.47 and 9.99. The structure of the pipeline explains this. Under a translation the coarse composite already contains an exact, undistorted copy of the object, and refinement only has to harmonise its boundary. Under a rotation or a resizing the object in the composite has been resampled, and refinement must restore detail that resampling degraded, at a pose that the attention transplant does not know about (§5.4.2).

The failure is concentrated in hard rotations and hard resizings. Subject consistency falls across the tertiles, 0.9417, 0.9274 and 0.8650, and the keypoint error rises from 4.86 to 6.98 and 13.51 pixels; most of the loss occurs in the last tertile. Severity alone does not explain it, however (Figure 7.4). From the easy to the hard tertile the keypoint error grows by a factor of 1.4 for translations, from 3.29 to 4.75 pixels, but by 3.3 for rotations, from 4.73 to 15.84, and by 2.9 for resizings, from 5.68 to 16.50: a hard translation is easier than an easy rotation. The two worst groups, hard rotations and hard resizings, contain 1,480 edits, 26% of the benchmark, and their keypoint

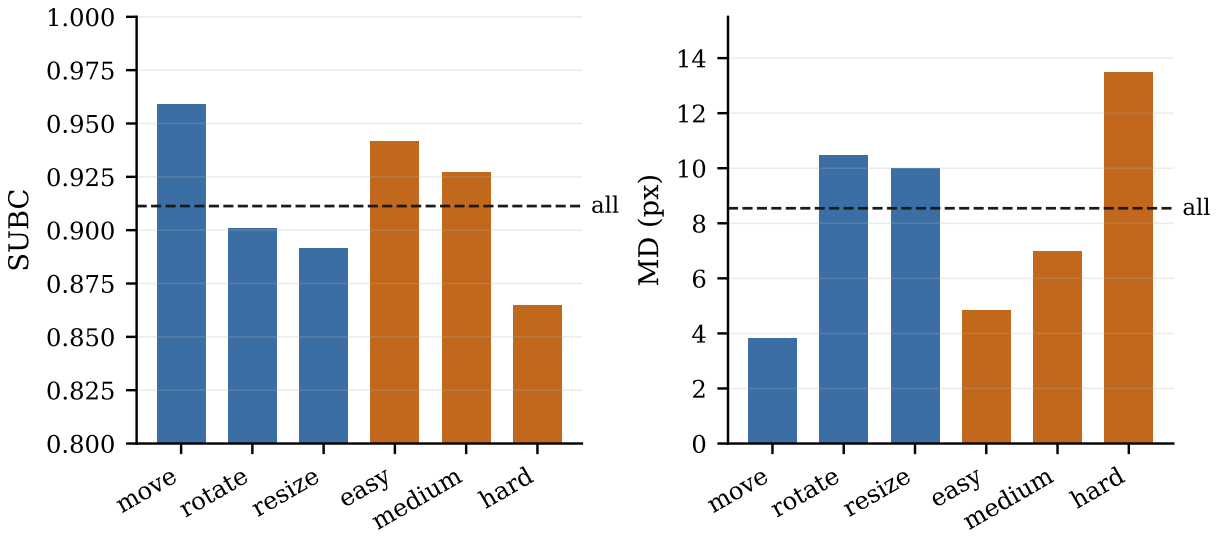


Figure 7.3: The reproduced baseline by edit type (first three bars of each panel) and by severity tertile (last three). The dashed line is the value over all edits.

Table 7.7: The reproduced baseline on the three severity tertiles of Eq. (7.2).

Group	n	SUBC↑	BGC↑	WRAP_E↓	MD↓	FID↓	FID_{dino}↓	KD↓
Easy	1,892	0.9417	0.9709	0.0462	4.86	57.59	772.07	0.191
Medium	1,892	0.9274	0.9661	0.0455	6.98	44.08	589.32	0.145
Hard	1,893	0.8650	0.9641	0.0503	13.51	42.17	533.77	0.167
All	5,677	0.9113	0.9670	0.0474	8.55	35.04	487.82	0.142

Mean distance as in Table 7.6. The realism columns are shown to support the argument of the text; they do not order the tertiles by quality.

error is roughly twice the value over all edits. These two groups are the target of the rest of the thesis.

Background consistency carries almost no information. Across all groups background consistency ranges over 0.007, and across the three edit types over 0.0004. The pipeline explains why. Outside the band of Eq. (5.9), refinement is a deterministic DDIM reconstruction of the composite, which there nearly coincides with the source, and the background term of TCA draws on the source background (§5.4.2); the background is therefore reproduced closely by construction. The metric remains useful as a guard against damage, but not as evidence of improvement.

The warping error and the keypoint error measure different things. Rotations have the lowest warping error of the three edit types, 0.0433, and the highest keypoint error, 10.47. The warping error measures agreement with the coarse composite, the keypoint error agreement with the analytic target position, and an output that stays close to the composite scores well on the first whether or not its geometry is where the second expects it. Across the severity tertiles the warping error moves only between 0.0455 and 0.0503, without order, so on its own it cannot separate a better method from a worse one.

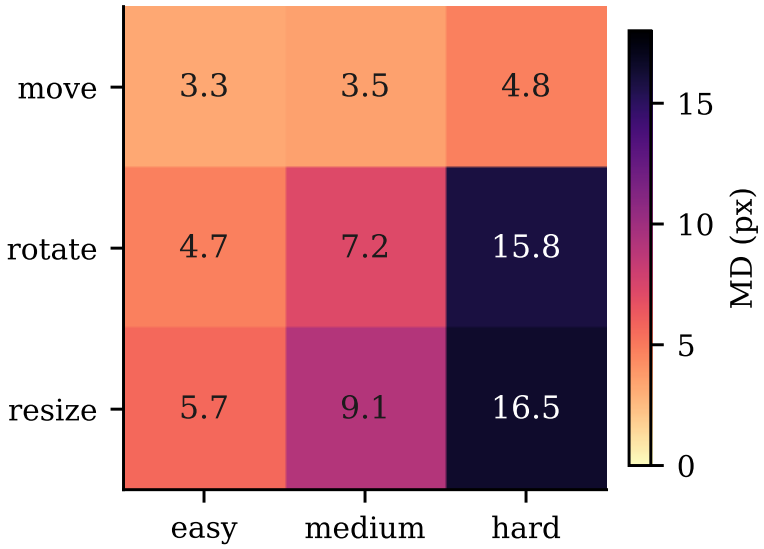


Figure 7.4: Mean distance, in pixels, in the nine groups of edit type by severity. The group sizes are given in Table 7.5.

Part of the rotation and resizing error is instrumental. The keypoint target is computed about the centre of mass of the mask, the transformation about the centre of its bounding box (§5.8). For a centre offset δ, an exactly executed rotation by ρ is charged a residual of

$$\|(\mathrm{I} - R_\rho)\delta\|_2 = 2|\sin(\rho/2)|\,\|\delta\|_2, \tag{7.3}$$

and an exactly executed scaling by s a residual of $|1-s|\,\|\delta\|_2$. At the largest rotation in the benchmark, 39°, the factor is 0.67; at the largest enlargement, $s = 2.88$, it is 1.88. An object whose centroid lies ten pixels from the centre of its bounding box, which is common for elongated or partly occluded objects, is therefore charged about seven and about nineteen pixels respectively before any generative error. The offset vanishes for translations and grows with the severity of the transformation, so part of the severity trend of the keypoint metric is a property of the instrument. Mean distance is therefore used to compare methods on the same group, never as an absolute measure of editing error.

The realism metrics cannot compare groups. The easy tertile scores a worse FID than the hard tertile, 57.59 against 42.17, at almost the same size, and the DINOv2 distance orders them in the same way; every subset scores worse than the whole (Figure 7.5). This is the fixed reference collection of §5.8 at work: each group repeats the 609 source images a different number of times and differs in size from the reference set, and the resulting mismatch dominates the value. The realism metrics are therefore used only to compare methods on the same group.

What the numbers look like. Figures 7.6 and 7.7 show FreeFine on four edits, from an easy translation to a severe enlargement. In every case the object is placed correctly and the scene is coherent; what changes is the fine structure of the object. The translated van keeps its shape, but refinement redraws its lettering and number plate. The rotated plane is cleaned of the holes and jagged edges of the rotated copy, and its details are redrawn. The severely reduced lynx loses the detail of its face and fur, and the legs of the severely enlarged bird fade. These are the losses that subject consistency and the keypoint error register, and they grow with the severity of the transformation.

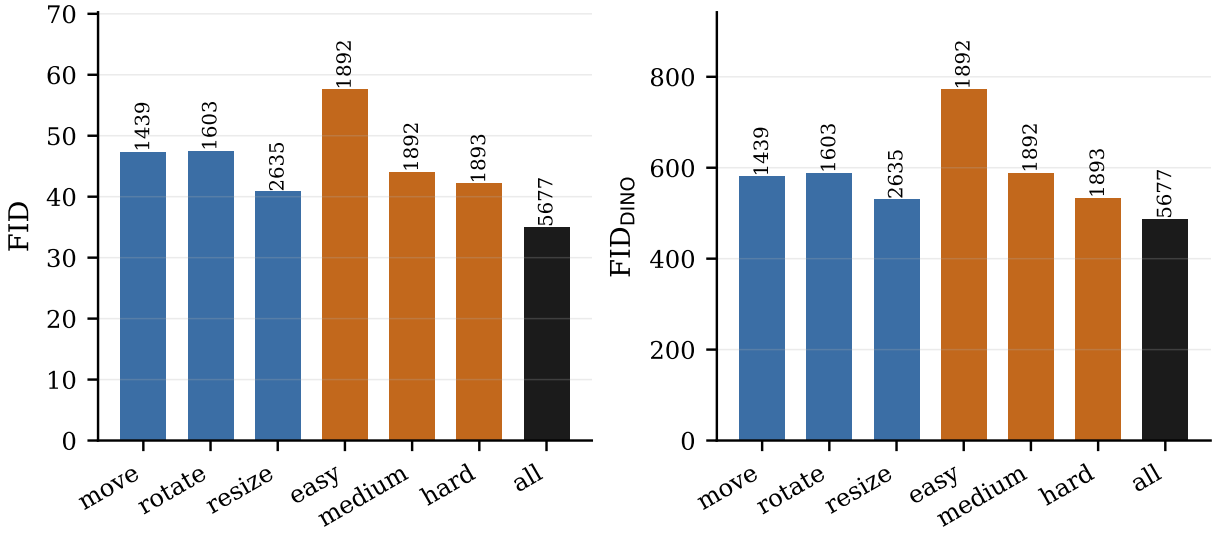


Figure 7.5: FID and the DINOv2 distance on the six groups and on all edits; the number above each bar is the size of the scored set. Every subset scores worse than the whole, and the easy tertile scores worst of all.

What comes next. The diagnosis fixes the target (hard rotations and hard resizings) and the terms on which methods are compared. Improvements must be measured on subject consistency and mean distance in those two groups, with translations as a control and the realism metrics compared within a group. Testing explanations on all 5,677 edits every time would be impractical, so the experiments of the following chapters run on a development subset built to represent these groups, under the protocol of Chapter 8. The first explanation to test is the simplest: that the released configuration is merely a poor setting of controls that FreeFine already exposes.

(a) Easy translation: 40 pixels upwards



(b) Hard rotation: 32 degrees

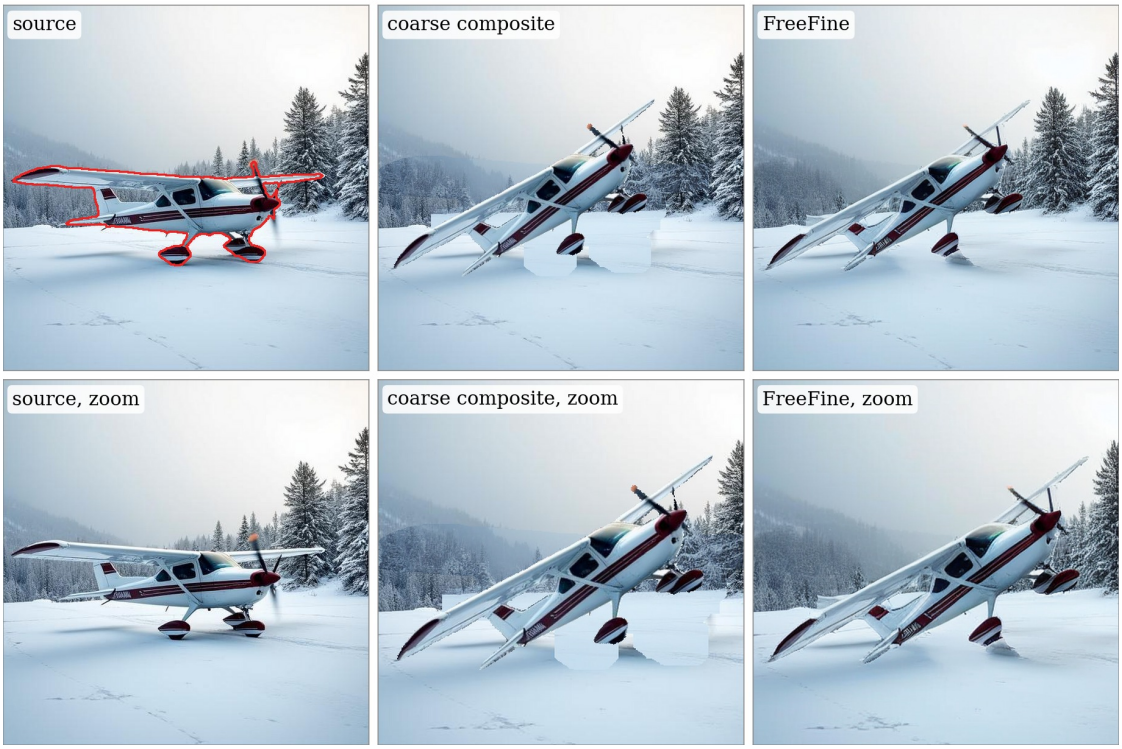


Figure 7.6: FreeFine on (a) an easy translation and (b) a hard rotation. In each part, the top row shows the source image (red: the object's mask), the coarse composite that FreeFine refines, and FreeFine's output; the bottom row enlarges the same region of each, marked by the yellow boxes (for the rotation, the whole frame).

(c) Severe reduction: to 0.49 of the original size



(d) Severe enlargement: to 1.95 of the original size (zoom on the legs)

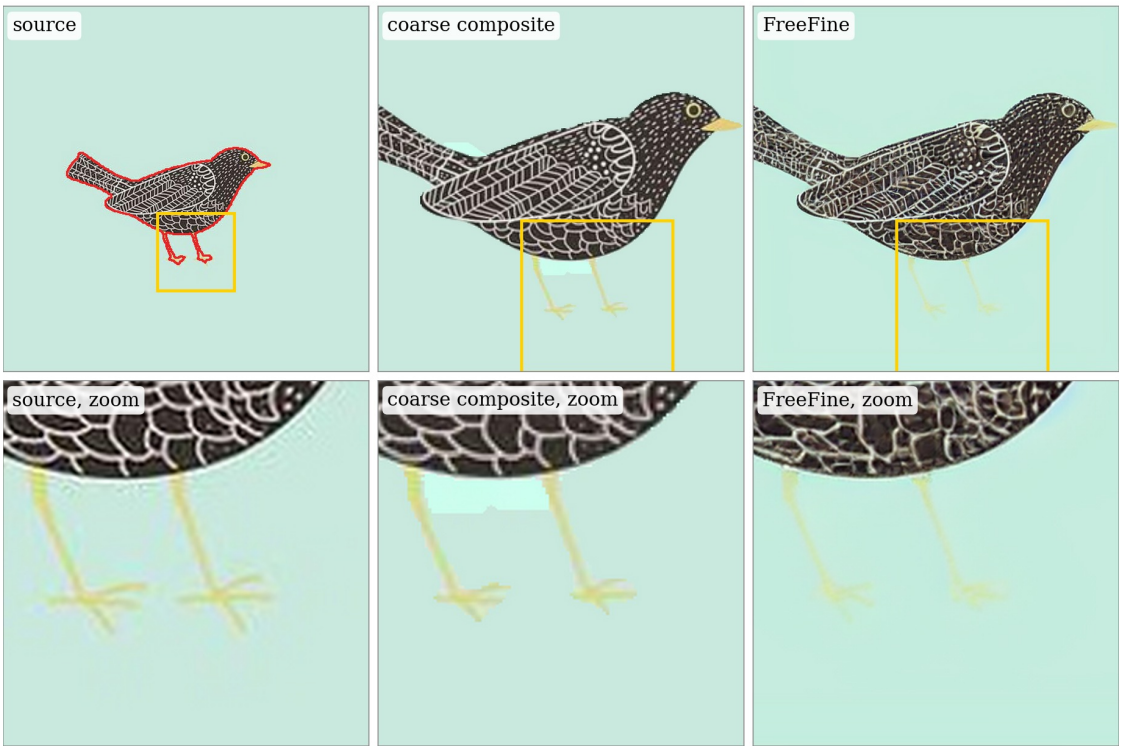


Figure 7.7: FreeFine on (c) a severe reduction and (d) a severe enlargement, laid out as in Figure 7.6. In the source image the yellow box marks the region that the transformation maps onto the enlarged region.

Chapter 8

Experimental Protocol

Every experiment from Chapter 9 onward follows one protocol, described here once. It fixes the edits on which methods are developed and checked, the groups on which they are judged, the controls that accompany every study, the treatment of the two stochastic metrics, and the rules by which results are read. The experiments of later chapters state only where they depart from it.

8.1 Development and held-out subsets

Testing an explanation means regenerating every edit with a modified pipeline and scoring the result. So that many variants can be compared under identical conditions, the experiments run on a development subset of 200 edits, and only the final method is evaluated on all 5,677 (Chapter 15).

The development subset. The development subset is drawn with a fixed seed (42) from the nine groups of edit type and severity tertile of Experiment 2 (Table 7.5): 22 edits are drawn uniformly from each group, which gives 198, and two more uniformly from the remaining edits. Table 8.1 gives its composition. Stratification gives the nine groups nearly equal weight and the three edit types equal representation, whereas the full benchmark is dominated by resizings (46% of its edits); the easy and medium groups, which serve as controls, are measured with the same precision as the two hard groups that are the target. The same seeded procedure re-derives the same 200 edits in every study, and the later studies also check the subset against its SHA-256 fingerprint (beginning `3d7c0172`) before using it.

The held-out subset. A second subset of 200 edits is drawn with a different seed (123) from the edits that are not in the development subset. It contains 66 translations, 68 rotations and 66 resizings, among them 24 hard rotations and 22 hard resizings. It is never used to develop a method; it serves only to check that a result obtained on the development subset holds on edits that played no part in obtaining it.

Table 8.1: Composition of the development subset of 200 edits, by edit type and severity tertile.

	easy	medium	hard	total
Move	23	22	22	67
Rotate	22	22	22	66
Resize	22	22	23	67
Total	67	66	67	200

Table 8.2: The four standard groups of the development subset and the metrics computed on each.

Group	n	Role	Metrics
Hard rotations	22	target	SUBC, BGC, WRAP_E, MD
Hard resizings	23	target	SUBC, BGC, WRAP_E, MD
Translations	67	control	SUBC, BGC, WRAP_E (and MD from Experiment 13)
Whole subset	200	realism	SUBC, BGC, WRAP_E, FID, FID_{dino}, KD (and MD from Experiment 16)

8.2 Groups

Four groups of the development subset are reported for every variant, each for a different reason (Table 8.2).

- **Hard rotations** (22 edits) and **hard resizings** (23 edits) are the two groups in which Experiment 2 located the failure. Every method is judged on them.

- **All translations** (67 edits) are a control. The baseline already handles translations well, so a method that damages them harms editing in general, whatever it does elsewhere.

- **The whole subset** (200 edits) is where background consistency and the three realism metrics are computed, since the realism metrics compare distributions and need every image.

Up to Experiment 12, mean distance is reported on the two hard groups only; from Experiment 13 on it is also reported on translations, and from Experiment 16 on the whole subset. Later chapters add groups where their question requires it (all resizings, for example, or resizings divided by the exact scale of the edit) and define them where they are used.

8.3 Controls that accompany every study

The switch-off check. Every modification is implemented behind a switch in the released code. Before any variant of a study is generated, three to five edits are regenerated with every switch off and compared pixel by pixel with the stored baseline images. In every study of this thesis the mean absolute difference was 0.0: the modified code reproduces the released computation exactly when its modifications are disabled, so a difference measured with a switch on is caused by the modification and not by the code that carries it.

Pairing. Every variant is generated from the same inpainted backgrounds and with the same seed as the baseline, and generation is deterministic (Experiment 1). Two variants are therefore paired image by image, and every difference between them is caused by the setting in which they differ.

Liveness. To confirm that a modification is active, the mean absolute difference between its images and the baseline images is measured on the first edits generated, on the 0–255 grey-level scale. A variant whose images equal the baseline, or whose metrics reproduce the baseline on every group to the last reported decimal, has not acted at all; in a deterministic pipeline this identifies a control that is exposed but ineffective, which Chapter 9 encounters twice.

8.4 The stochastic metrics

Mean distance. From Chapter 9 onward the random state of the keypoint metric is reset to a fixed seed at the start of every evaluation, so the same images always give the same value; two independent evaluations of the baseline agree to all four reported decimals. Seeding makes the metric repeatable, not paired: each variant is still scored on its own keypoints (§5.8). How far the metric moves between variants whose images barely differ is measured in Experiment 3 and used from then on to decide which differences in mean distance count.

Kernel distance. The kernel distance becomes repeatable only when the global random state of the evaluation is fixed as well, which was done in the later studies; every table states whether its kernel distance is seeded. Where it is not, repeated evaluations of the same baseline images give values between 0.1228 and 0.1285, so differences smaller than about 0.006 are not interpreted, and each table reports the baseline's kernel distance from the same evaluation as its variants.

The warping error. The earlier studies computed the warping error with a direct implementation of Eq. (5.11), using the coarse composite and target mask stored with the benchmark; the later studies use the official implementation. The two give the same values on the same images: 0.0488 for the baseline on the development subset in both.

The realism metrics. FID, the DINOv2 distance and the kernel distance are computed against the benchmark's full directory of source images. On 200 edits they are much larger than on the full benchmark (the baseline scores an FID of 132.28 on the development subset against 35.04 on all edits) and, as Experiment 2 showed, they are comparable only between methods scored on the same group.

8.5 Reading the results

Table layout. Every study reports its variants in one table with two blocks. The first gives subject consistency, the warping error and mean distance on the two hard groups; the second gives subject

consistency and the warping error on translations, and subject consistency, background consistency, the warping error and the three realism metrics on the whole subset. The released baseline is the first row, arrows mark the direction in which each metric improves, and background consistency on the hard groups, which rarely moves, is given in a note.

Reporting rules. Four rules, which follow from the properties of the metrics (§5.8) and from Experiments 1 and 2, are applied throughout.

1. Every result is stated together with the group it holds on and the size of that group.

2. The two stochastic metrics are compared only between variants scored within the same seeded evaluation, never against a value obtained elsewhere.

3. The realism metrics are compared only between variants scored on the same group.

4. A metric that a method improves by construction is reported but not used as evidence. This applies in particular to the warping error, and in part to subject consistency, of any method that copies the coarse composite into its output.

Differences and percentages quoted in the text are computed from the unrounded values. They can therefore differ by one unit in the last digit from what the rounded entries of a table give.

The criterion for an improvement. What counts as an improvement is derived from data rather than fixed in advance. The threshold for subject consistency is set at the end of Chapter 9 from the spread that re-configuring the released pipeline produces, and each later experiment states the criterion it is judged by.

8.6 Provenance

All generation uses the public FreeFine repository with the changes of Experiment 1; the later studies pin it at commit `4c9fdb97`. In both cases the switch-off check reproduces the stored baseline exactly, so every study in this thesis runs the same computation. All scoring uses the authors' evaluation code with the two changes of Experiment 1. The later evaluations also record, for each group, the number of edits it should contain and whether all of them were scored; every such group is complete.

Chapter 9

Hypothesis 1: Tuning the Released Pipeline

The simplest explanation of the failure on hard rotations and resizings is that nothing is missing from FreeFine at all: the released configuration is one setting of controls that the method already exposes, and a better setting may exist. This chapter tests that explanation before any mechanism is added.

> **Hypothesis 1.** Some setting of the controls that the released pipeline exposes, or of the quality of the input it refines, raises subject consistency on a hard group clearly above the baseline without a loss elsewhere.

Five experiments vary one factor at a time around the released configuration (Table 9.1): the refinement strength (Experiment 3), the attention layers modulated by TCA (Experiment 4), guidance with the object label (Experiment 5), the floor of the TCA schedule (Experiment 6), and the resampling that produces the coarse composite (Experiment 7). Together they also measure how far the metrics move under changes that add no new information to the system, which fixes the threshold that the mechanisms of the following chapters must clear.

9.1 Experiment 3: Refinement strength

9.1.1 Change

The start step s is the one documented control that acts in the released configuration, and it is easily misread as the step at which the attention transplant begins. The refinement loop reads otherwise. The composite is inverted up to step s of the 50-step grid, and only the last $50 - s$ steps are denoised (§5.4.1), so refinement enters at the timestep

$$t_s = 981 - 20\,s. \tag{9.1}$$

The start step is therefore the strength of the partial denoising of §3.4. One parameter fixes three quantities that cannot be varied independently: the noise level at which refinement starts, and with it how much of the composite the model may rewrite; the number of denoising steps; and the window over which the TCA weight γ_i falls from one (Eq. (5.6)). Table 9.2 lists them for the values tested. A small s lets refinement rewrite the composite heavily; a large s leaves it almost untouched.

Table 9.1: Settings varied in this chapter, one at a time around the released configuration.

Setting	Symbol	Released	Values tested	Experiment
Refinement strength	s	35	$25, 30, 40, 45$	3
TCA layer range	ℓ_0	10	$4, 8, 12$	4
Guidance with the object label	w	none	$7.5, 10$	5
Floor of the TCA schedule	e	0	$0.25, 0.5$, by severity[a]	6
Warp of the coarse composite		bilinear	anti-aliased[b]	7

[a] $e = 0$, 0.25 and 0.5 for the easy, medium and hard tertiles.
[b] Also combined with the severity-dependent floor.

Table 9.2: What the start step sets. Signal and noise coefficients are those of the SD-1.5 schedule at the entry timestep.

s	Refined steps	t_s	$\sqrt{\bar{\alpha}_{t_s}}$	$\sqrt{1-\bar{\alpha}_{t_s}}$
25	25	481	0.550	0.835
30	20	381	0.675	0.738
35	15	281	0.789	0.615
40	10	181	0.885	0.466
45	5	81	0.958	0.288

9.1.2 Results

Table 9.3 and Figure 9.1 report the sweep.

Every consistency and pixel metric is monotone in s. As refinement weakens, subject consistency rises in every group (from 0.8497 to 0.8529 on hard rotations, from 0.8319 to 0.8414 on hard resizings, from 0.9542 to 0.9612 on translations), and the warping error falls in every group, from 0.0540 to 0.0452 on the whole subset. The realism metrics move the other way: the DINOv2 distance rises from 1636.08 to 1643.70, and background consistency falls from 0.9667 to 0.9641. Mean distance has no monotone trend. On hard rotations it reads 15.25, 15.91, 14.01, 15.01 and 15.09, so the released value is lower than both its neighbours, by 1.0 and 1.9 pixels; on hard resizings it reads 21.13, 20.70, 19.47, 20.16 and 18.09.

9.1.3 Outcome

Refinement strength is a trade-off, not a lever. As s approaches 50 the entry noise vanishes, refinement has nothing left to rewrite, and the output approaches the coarse composite itself. The consistency metrics compare the output with the source object and with that composite, so they reward the approach; the realism metrics penalise the seams and the resampling of the composite, so they punish it. The start step moves the output along a single axis between regenerating and copying, and it trades one family of metrics against the other. The largest gain in subject consistency, $+0.0023$ on hard resizings at $s = 45$, is bought with a DINOv2 distance 6.9 units higher; on hard rotations the best gain is $+0.0006$. The released value is an interior compromise on this axis, not a mistake.

Table 9.3: The start-step sweep; the released value is $s = 35$. All rows come from one evaluation.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
s	SUBC↑	WRAP__E↓	MD↓	SUBC↑	WRAP__E↓	MD↓
25	0.8497	0.0459	15.25	0.8319	0.0658	21.13
30	0.8519	0.0437	15.91	0.8376	0.0619	20.70
35	0.8523	0.0416	14.01	0.8391	0.0585	19.47
40	0.8528	0.0398	15.01	0.8413	0.0555	20.16
45	0.8529	0.0384	15.09	0.8414	0.0535	18.09

	Translations ($n = 67$)		Whole subset ($n = 200$)					
s	SUBC	WRAP__E	SUBC	BGC	WRAP__E	FID	FID_{dino}	KD
25	0.9542	0.0534	0.9113	0.9667	0.0540	132.57	1636.08	0.1270
30	0.9571	0.0512	0.9139	0.9663	0.0513	132.73	1636.61	0.1264
35	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1285
40	0.9605	0.0480	0.9166	0.9650	0.0468	133.09	1640.26	0.1347
45	0.9612	0.0468	0.9174	0.9641	0.0452	133.58	1643.70	0.1397

Background consistency on the hard groups: 0.9666–0.9681 (rotations) and 0.9618–0.9637 (resizings). Kernel distance is not seeded.

What mean distance can resolve. A monotone effect of refinement strength could not produce the shapes of the mean-distance rows. With 22 and 23 edits, and each variant scored on its own keypoints (§5.8), variants whose images differ only slightly already differ by one to two pixels on the hard groups. From here on a change in mean distance of that size on a single variant is not treated as an effect; a reduction counts when it is clearly larger, or when it grows consistently with the strength of a mechanism.

Next. The start step is the only documented control that acts in the released configuration. The other two are examined next.

9.2 Experiment 4: The layer range of TCA

9.2.1 Change

The released script exposes a first layer ℓ_0, and the attention controller is built to modulate the transformer blocks $\ell_0, \ldots, 15$ of Table 3.1. Setting $\ell_0 = 8$ and $\ell_0 = 12$ reproduced the baseline on every group to all four decimals (mean distance on hard resizings, for example, stayed at 19.4713), which in a deterministic pipeline scored with a seeded metric means that the setting never reached the computation. The model code confirms it: after the controller is built, it assigns the blocks $\{10, \ldots, 15\}$ at five separate places, so the configured value is discarded. Those five assignments were replaced by the configured value, which made the control act.

With the control live, three ranges were tested. TCA acts only on decoder blocks, so $\ell_0 = 4$ extends it to all nine decoder blocks, adding the three at 16×16; $\ell_0 = 8$ adds the last two blocks at 16×16;

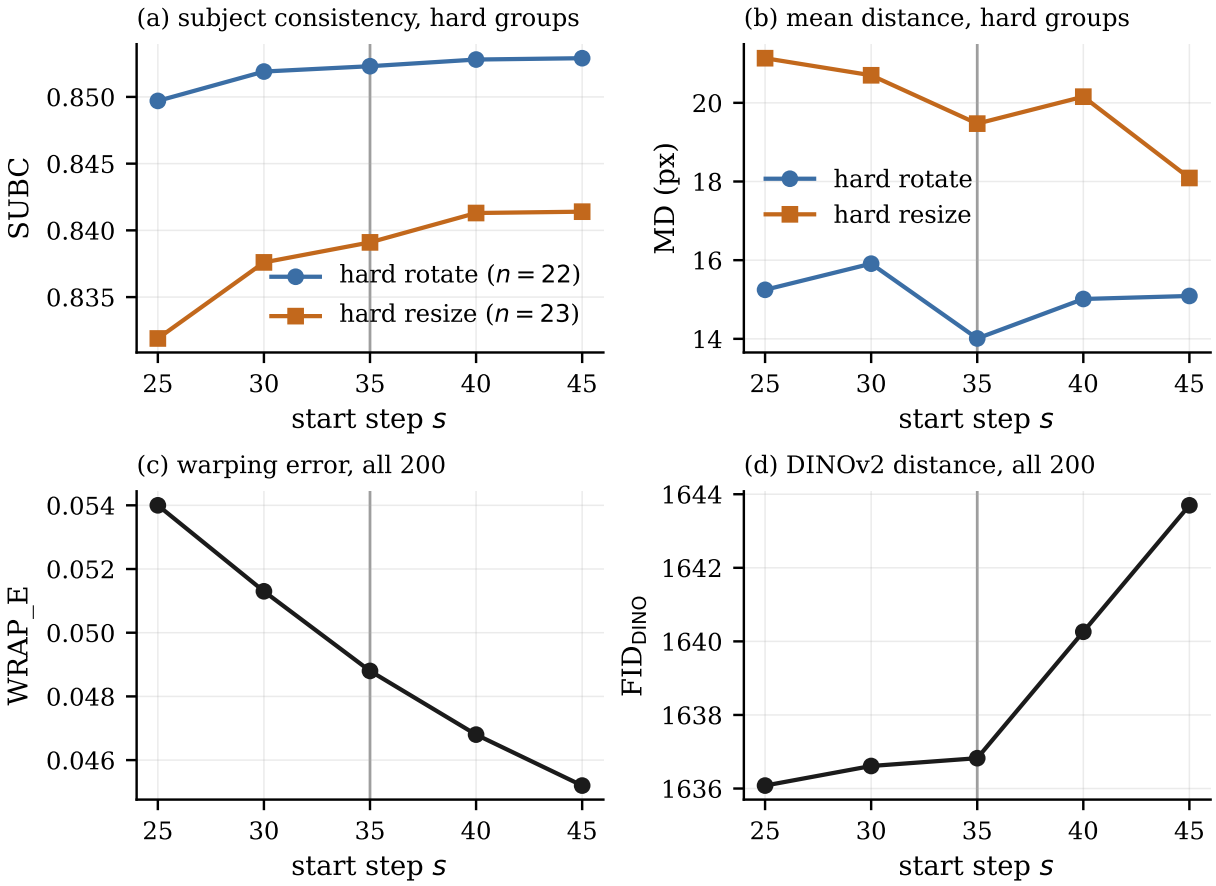


Figure 9.1: The start-step sweep. Subject consistency rises and the warping error falls monotonically as refinement weakens, while the DINOv2 distance rises; mean distance on the hard groups is not monotone. The vertical line marks the released value.

and $\ell_0 = 12$ removes the first two blocks at 32×32. The released range, blocks 10–15, covers the six decoder blocks at 32×32 and 64×64.

9.2.2 Results

Table 9.4 reports the three ranges. Subject consistency stays within 0.0006 of the baseline on every group except hard resizings, where all three ranges lower it, by 0.0008 to 0.0013. Mean distance on hard resizings worsens by 2.1 to 3.0 pixels for every range, and on hard rotations by 0.3 to 1.3 pixels.

9.2.3 Outcome

The control acts but does nothing useful: adding coarser decoder blocks and removing finer ones fail alike. The result follows from what TCA is (§5.4.2). The layer range decides *where* in the network the source object's features are transplanted, not whether they correspond to the target pose, and that correspondence is unaffected by including more or fewer layers. The released range is at least as good as the alternatives and is kept.

Table 9.4: The layer range of TCA, once the control acts. The released range starts at block 10.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
ℓ_0	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
10 (released)	0.8523	0.0416	14.01	0.8391	0.0585	19.47
4	0.8524	0.0425	15.30	0.8378	0.0587	21.69
8	0.8522	0.0424	14.33	0.8382	0.0586	21.61
12	0.8517	0.0412	14.68	0.8383	0.0579	22.45

	Translations ($n = 67$)		Whole subset ($n = 200$)					
ℓ_0	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
10 (released)	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1285
4	0.9587	0.0497	0.9152	0.9658	0.0492	132.41	1636.90	0.1264
8	0.9587	0.0496	0.9152	0.9658	0.0491	132.50	1637.17	0.1303
12	0.9590	0.0494	0.9152	0.9650	0.0486	132.82	1637.51	0.1276

Kernel distance is not seeded.

9.3 Experiment 5: Guidance with the object label

9.3.1 Change

The third documented control is the guidance scale w of Content-specified Generation. Scales of 5 and 10 also reproduced the baseline on every group to all four decimals. The cause is the prompt: the released script sets the object label to the empty string, and the unconditional prediction is computed from the empty string as well. Both predictions therefore receive the same conditioning and are equal, and by the first property of §3.7 the guidance term of Eq. (5.8) is identically zero:

$$\hat{\varepsilon} = \varepsilon_{\varnothing} + w\,(\varepsilon_{\varnothing} - \varepsilon_{\varnothing}) \odot m_{\text{cfg}} = \varepsilon_{\varnothing} \qquad \text{for every } w. \tag{9.2}$$

Guidance becomes active as soon as it has something to guide towards. GeoBench provides a category label for every object, and passing it as the prompt p restores the correction term: the prediction is pushed along $\varepsilon_p - \varepsilon_{\varnothing}$, the direction in which the model's prediction changes when it is told what the object is. Two scales were tested, the released $w = 7.5$ and $w = 10$. Where the correction acts is fixed by Eq. (5.9): the guided region is the band around the transformed object together with the area it vacated, and it excludes the object itself. Guidance can therefore reach the object only indirectly, through the context in which the object is refined.

9.3.2 Results

Table 9.5 reports the two scales. On hard resizings mean distance falls from 19.47 to 18.38 at $w = 7.5$ and to 17.75 at $w = 10$, a reduction of 1.73 pixels or 8.9% that grows with the scale. Nothing else moves: subject consistency stays within 0.0006 of the baseline on every group, the warping error within 0.0002, and the DINOv2 distance rises by about one unit. On hard rotations mean distance rises, by 0.54 and 0.68 pixels.

Table 9.5: Content-specified Generation with the object label as the prompt.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Baseline (empty label)	0.8523	0.0416	14.01	0.8391	0.0585	19.47
Object label, $w = 7.5$	0.8528	0.0416	14.55	0.8390	0.0586	18.38
Object label, $w = 10$	0.8529	0.0416	14.69	0.8391	0.0587	17.75

	Translations ($n = 67$)		Whole subset ($n = 200$)					
	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
Baseline	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1285
$w = 7.5$	0.9588	0.0496	0.9154	0.9656	0.0489	132.35	1637.84	0.1273
$w = 10$	0.9587	0.0497	0.9154	0.9657	0.0490	132.51	1637.89	0.1326

Kernel distance is not seeded.

9.3.3 Outcome

The first live lever. This is the first change that lowers mean distance on a hard group by clearly more than the variation of the metric, and it does so with a dose response and without cost elsewhere. That subject consistency and the warping error do not move is what Eq. (5.9) predicts: the object lies outside the guided region, and only its surroundings are steered.

Why resizings and not rotations. The most likely explanation is the shape of the guided region. A rotation about the bounding-box centre leaves the source and target masks largely overlapping, so the band is mostly a thin ring around the object. A resizing leaves a wide ring between the object's old and new outlines, and in that ring structure at the original scale must be replaced by coherent content; residual structure at the wrong scale is exactly the kind of content that a correspondence-based metric can match to. Telling the model what the object is steers the content of that ring towards the object's category, and the keypoints located there move closer to their targets.

Next. The effect is specific to resizing and does not reach subject consistency, so it does not confirm Hypothesis 1. It is nevertheless the one positive result of the chapter, and it returns in the combined models of Chapter 11 and in the guidance studies of Chapter 12. The remaining two experiments change the schedule of the transplant and the input to refinement.

9.4 Experiment 6: The floor of the TCA schedule

9.4.1 Change

The implementation ramps the TCA weight from one towards a floor e (Eq. (5.6)), which the released configuration sets to zero. With $e > 0$ the transplant is never fully released: in the last refined steps the edit branch still takes a fraction of about e of its attention output from the source object (Figure 9.2). The idea behind raising the floor is that consistency is lost during the release phase, when the edit is left to its own self-attention, and that keeping part of the transplant to the end preserves more of the

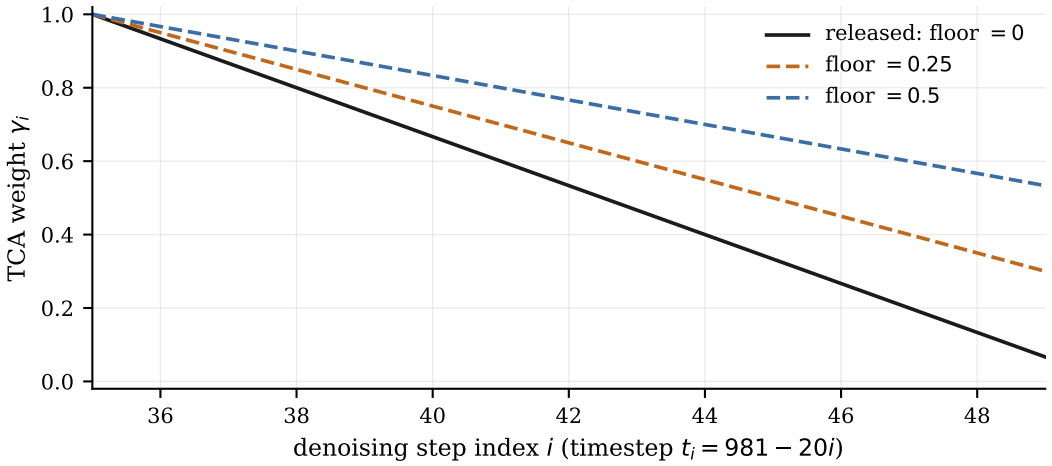


Figure 9.2: The TCA weight γ_i over the fifteen refined steps for the released floor $e = 0$ and the two raised floors.

Table 9.6: The floor of the TCA schedule. The severity-dependent floor coincides with $e = 0.5$ on the hard groups.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
Floor	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
$e = 0$ (released)	0.8523	0.0416	14.01	0.8391	0.0585	19.47
$e = 0.25$	0.8524	0.0423	14.74	0.8388	0.0591	22.06
$e = 0.5$	0.8522	0.0434	14.96	0.8380	0.0600	24.45
By severity	0.8522	0.0434	14.96	0.8380	0.0600	24.45

	Translations ($n = 67$)		Whole subset ($n = 200$)					
Floor	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
$e = 0$ (released)	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1270
$e = 0.25$	0.9588	0.0496	0.9155	0.9660	0.0492	132.31	1637.94	0.1250
$e = 0.5$	0.9582	0.0500	0.9152	0.9666	0.0498	132.52	1638.50	0.1307
By severity	0.9583	0.0499	0.9151	0.9661	0.0494	132.55	1638.05	0.1247

Kernel distance is not seeded.

source object. Floors of 0.25 and 0.5 were tested, and a severity-dependent variant that sets $e = 0$, 0.25 and 0.5 for the easy, medium and hard tertiles, on the reasoning that only severe edits need the extra anchoring. On the two hard groups this variant coincides with $e = 0.5$.

9.4.2 Results

Table 9.6 reports the floors. Subject consistency changes by between -0.0011 and $+0.0001$ on every group. Mean distance on hard resizings worsens monotonically with the floor, from 19.47 to 22.06 and 24.45, an increase of 26% at $e = 0.5$; on hard rotations it worsens by 0.73 and 0.95 pixels. On the hard groups the severity-dependent floor reproduces $e = 0.5$ to every decimal, as it should.

9.4.3 Outcome

The idea behind the floor is refuted: keeping the transplant to the end does not preserve more of the object. The direction of the damage shows why. The reference branch of TCA reconstructs the *source* image, in which the object is at its original size and orientation, and a raised floor keeps injecting features of that object into the target region until the last step. Raising the floor strengthens the part of the system that does not know the transformation, and the metric that measures the transformation degrades, most of all under resizing. The released schedule, which releases the transplant almost completely, is the better one. Every change so far acted on the refinement; the last experiment acts on its input.

9.5 Experiment 7: An anti-aliased coarse composite

9.5.1 Change

The coarse composite is produced by an affine warp with bilinear interpolation (§5.3). If part of the failure is inherited from resampling artefacts in the composite, a better warp should reduce it.

Why a bilinear warp aliases. An affine warp computes each output pixel by evaluating an interpolated source at the pre-image of the pixel centre,

$$I'(\mathbf{u}) = \sum_{\mathbf{x}} I(\mathbf{x})\, h(A^{-1}(\mathbf{u} - \mathbf{b}) - \mathbf{x}), \tag{9.3}$$

where $(A, \mathbf{b})$ is the affine map and h the interpolation kernel; for bilinear interpolation $h(x, y) = \mathrm{tri}(x)\,\mathrm{tri}(y)$ with $\mathrm{tri}(t) = \max(0, 1 - |t|)$. The kernel is two source pixels wide whatever the transformation [52]. Under a reduction by $s < 1$ the output samples the source at s samples per source pixel, so source content above $s/2$ cycles per pixel must be removed before sampling. The fixed-width kernel does not remove it, and it folds back into the output as spurious low-frequency structure; at the strongest reduction of the benchmark, $s = 0.4$, everything between 0.2 and 0.5 cycles per pixel is affected. Under enlargement and rotation there is no reduction, and the weakness of the bilinear kernel is instead that it softens fine detail.

The anti-aliased warp. The replacement upsamples the source by a factor of two with a Lanczos kernel [53], applies the same affine map in the doubled coordinate frame with Lanczos interpolation, and reduces the result back by area averaging. If the map is $\mathbf{u} = A\mathbf{x} + \mathbf{b}$, doubling both coordinate frames gives $2\mathbf{u} = A(2\mathbf{x}) + 2\mathbf{b}$, so only the translation is scaled:

$$M_2 = [\, A \mid 2\mathbf{b} \,]. \tag{9.4}$$

The area reduction averages the two-by-two samples inside each output pixel, a box filter of exactly one output pixel. In source coordinates the doubled grid samples at rate $2s$, so the construction removes aliasing up to s cycles per pixel and attenuates the band between $s/2$ and s; below $s = 0.5$ some aliasing remains. The mask is still warped with nearest-neighbour interpolation, so the geometry of the composite is unchanged and only the texture inside the object differs. By this analysis the replacement should matter most for strong reductions and least for rotations. It was tested alone and combined with the severity-dependent floor of Experiment 6.

Table 9.7: The anti-aliased warp, alone and combined with the severity-dependent floor. The warping error is given against the composite the variant refined (own) and against the benchmark's stored composite (bench).

	Hard rotations ($n = 22$)				Hard resizings ($n = 23$)			
	SUBC↑	WRAP_E↓		MD↓	SUBC↑	WRAP_E↓		MD↓
		own	bench			own	bench	
Baseline	0.8523	n/a	0.0416	14.01	0.8391	n/a	0.0585	19.47
Anti-aliased	0.8537	0.0447	0.0433	14.54	0.8403	0.0573	0.0587	19.67
Anti-aliased + floor[a]	0.8536	0.0463	0.0450	15.02	0.8388	0.0590	0.0604	20.30

	Translations ($n = 67$)			Whole subset ($n = 200$)					
	SUBC	WRAP_E		SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
		own	bench			bench			
Baseline	0.9590	n/a	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1270
Anti-aliased	0.9592	0.0454	0.0480	0.9156	0.9654	0.0490	132.36	1637.27	0.1297
Anti-aliased + floor[a]	0.9586	0.0459	0.0485	0.9153	0.9659	0.0496	132.63	1638.42	0.1255

[a] Combined with the severity-dependent floor of Experiment 6: every edit of the hard groups runs with $e = 0.5$; in the translations and the whole subset the floor varies with the severity of each edit.
On the whole subset the warping error against the variant's own composite is 0.0486 for the anti-aliased warp and 0.0492 in combination with the floor. Kernel distance is not seeded.

9.5.2 Results

Because the variant refines a different composite, its warping error is given against both references: the anti-aliased composite it was generated from (own) and the composite stored with the benchmark (bench), which every other row of this chapter uses.

Table 9.7 reports the variants. The anti-aliased warp raises subject consistency by 0.0014 on hard rotations and 0.0012 on hard resizings, the largest gains of the chapter, and worsens mean distance by 0.53 and 0.20 pixels. The warping error tells no single story. On translations the variant is closer to either composite than the baseline is, 0.0454 and 0.0480 against 0.0495; on hard rotations it is further from both, 0.0447 and 0.0433 against 0.0416; on hard resizings it is closer to its own composite and level with the benchmark's. Combined with the floor, it behaves mostly like the floor alone, except that its mean distance on hard resizings, 20.30, is far below the 24.45 of the floor with the bilinear composite.

9.5.3 Outcome

A better composite reaches the output only in small part. The reason lies in what refinement does to the object: the composite is inverted to $t = 281$ and denoised again, and during that denoising the attention of the target region is taken, with a weight that starts at one, from the source object in the reference branch (Eq. (5.5)). The texture of the refined object is therefore re-synthesised largely from source features rather than copied from the composite, and the difference between the two composites mostly does not survive. The anti-aliased warp is correct image processing, and it gives the largest consistency gains of the chapter, but it does not change the failure; the released warp is kept for all later experiments, so that they remain comparable with the reproduced baseline. The interaction with the floor on hard resizings is recorded but not pursued, since neither component improves the target

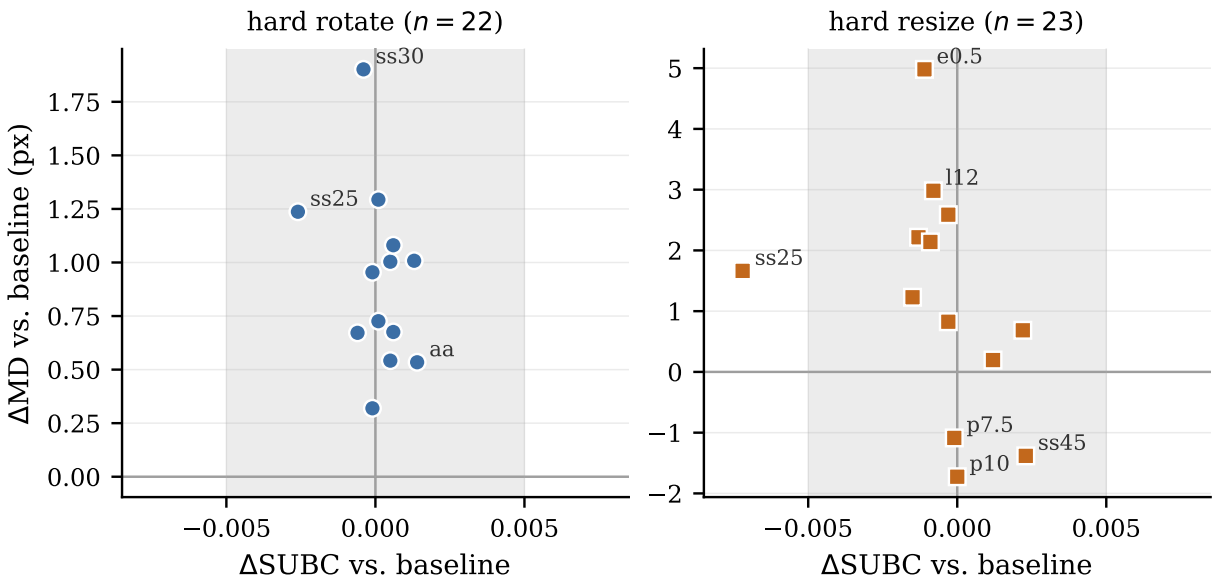


Figure 9.3: Every variant of Experiments 3–7 against the baseline on the two hard groups: change in subject consistency against change in mean distance, where lower is better. The shaded band marks changes of subject consistency within ±0.005. Labels: s start step, ℓ first layer, p object label with guidance scale, e floor, aa anti-aliased warp. The severity-dependent floor coincides with $e = 0.5$ on these groups and is not drawn separately.

metrics on its own.

9.6 The configuration space as a whole

Figure 9.3 places every variant of Experiments 3–7 relative to the baseline on the two hard groups, and Table 9.8 summarises them by setting.

Across the fourteen active variants, subject consistency on the hard groups moves between −0.0026 and +0.0014 on rotations and between −0.0072 and +0.0023 on resizings. Hypothesis 1 is rejected: no setting of the released controls, and no improvement of the composite's resampling, raises subject consistency on a hard group appreciably.

The same spread fixes the threshold for the rest of the thesis. Re-configuring the released pipeline raises hard-group subject consistency by at most 0.0023; a mechanism counts as improving subject consistency only if it raises it on a hard group by more than 0.005, about twice that amount, without lowering it on translations or lowering background consistency. A mechanism that clears this threshold does something that re-configuring the pipeline cannot. For mean distance, the rule of Experiment 3 applies.

The settings tested here therefore do not explain the failure. Two explanations remain. Either the refinement lacks information that it would need (what the object is, or how it was transformed), which Chapter 10 tests by supplying that information; or the information is present in its input, the coarse composite, and is lost during refinement, which Chapter 11 examines.

Table 9.8: The best change in hard-group subject consistency, and the range of changes in hard-group mean distance, produced by each setting relative to the baseline.

Setting	Acts as released	Best ΔSUBC	ΔMD, rotations	ΔMD, resizings
Guidance scale, empty label	no	0	0	0
Layer range	no	0	0	0
Layer range, made live	n/a	+0.0001	+0.32 to +1.29	+2.14 to +2.98
Start step	yes	+0.0023	+1.00 to +1.90	−1.38 to +1.66
Guidance, object label	n/a	+0.0006	+0.54 to +0.68	−1.73 to −1.09
Floor of the TCA schedule	yes	+0.0001	+0.73 to +0.95	+2.59 to +4.98
Anti-aliased warp[a]	n/a	+0.0014	+0.53 to +1.01	+0.20 to +0.83

"Acts as released" records whether the setting reaches the computation in the released configuration; "n/a" marks settings that are not part of it.

[a] Including its combination with the severity-dependent floor.

Chapter 10

Hypotheses 2 and 3: Supplying Information the Model Lacks

Chapter 9 showed that no setting of FreeFine's controls repairs the failure on hard rotations and resizings. This chapter asks whether the refinement stage lacks the information it needs. Refinement receives the coarse composite, in which the object already stands at its target pose, the source image, whose features TCA transplants, and an empty prompt (§5.4). Two things are never given to it: an explicit description of *what* the object is, since the network knows the object only through the features it copies; and *how* the object was transformed, since the affine map is used once to build the composite and then discarded, and TCA chooses among source tokens by feature similarity alone (§5.4.2).

> **Hypothesis 2 (identity).** Supplying an explicit representation of the source object's identity at every refinement step raises subject consistency on a hard group by more than 0.005.

> **Hypothesis 3 (geometry).** Supplying the exact geometry of the edit raises subject consistency on a hard group by more than 0.005, or clearly lowers mean distance on a hard group, without a loss on translations or on the background.

A frozen U-Net offers three places where information from outside can enter (Figure 10.1). *Cross-attention tokens* can be added next to the text tokens; they carry whatever they encode, but every image position reads the same tokens. *Residual feature maps* computed by an auxiliary network can be added to the skip connections; they are aligned with the image and can carry shape and pose directly. And the *self-attention* keys and values that a query reads can be replaced or restricted, which is how TCA already works. The four experiments of this chapter use all three: identity through image tokens (Experiment 8), geometry through image tokens (Experiment 9), geometry through residual feature maps (Experiment 10), and geometry inside the attention transplant (Experiment 11).

10.1 Experiment 8: Identity through image tokens

10.1.1 Change

The established way to tell a frozen diffusion model what an object looks like is to show it an image of the object through an image-prompt adapter.

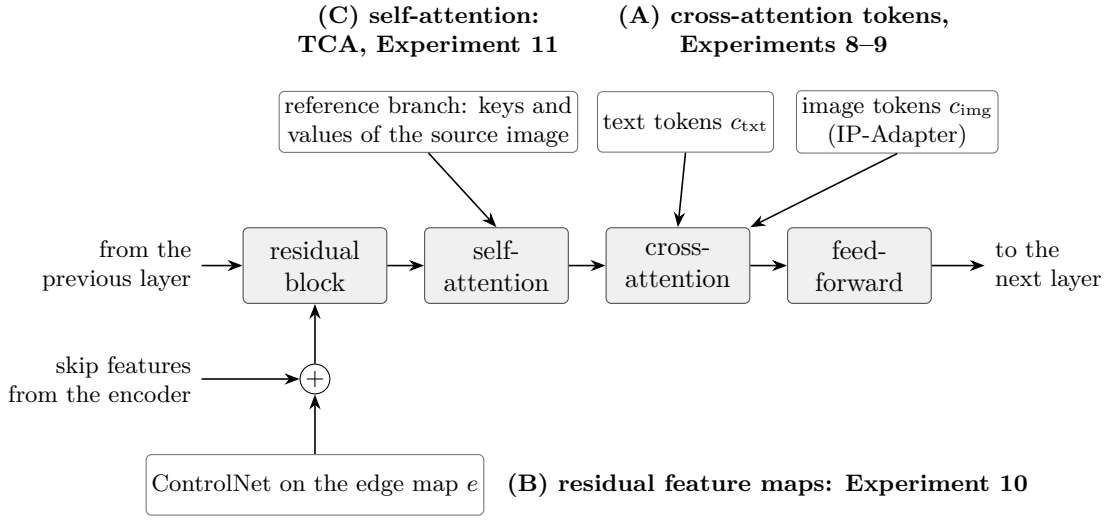


Figure 10.1: One decoder layer of the Stable Diffusion U-Net (§3.5) and the three places where information from outside can enter while the network stays frozen: (A) tokens read by cross-attention, (B) feature maps added to the skip connections, (C) the keys and values of self-attention, which TCA draws from the reference branch.

IP-Adapter. IP-Adapter [38] encodes an image with a frozen CLIP image encoder [10] and projects the embedding into a few tokens c_{img} of the same width as the text tokens. Appending them to the text tokens would make the two compete within one softmax, so every cross-attention layer instead receives a second, *decoupled* attention over the image tokens, with its own key and value projections, whose output is added to that of the text attention:

$$Z = \text{Attn}(Q,\, c_{\text{txt}}W_k,\, c_{\text{txt}}W_v) + \lambda\ \text{Attn}(Q,\, c_{\text{img}}W'_k,\, c_{\text{img}}W'_v), \tag{10.1}$$

where Q are the queries of the layer, c_{txt} the text tokens with the frozen projections W_k and W_v, and W'_k, W'_v the new projections. Only W'_k, W'_v and the projection of the embedding are trained: about 22 million parameters, on about ten million image-text pairs, with the U-Net frozen. The scale λ sets the strength of the image prompt; $\lambda = 0$ recovers the original model. The checkpoint for SD-1.5 used here encodes the image with CLIP ViT-H/14 and projects its pooled 1024-dimensional embedding linearly to four tokens of width 768.

Integration into FreeFine. Three decisions fix how the adapter enters the four-branch refinement of §5.4.1.

- **Only the two edit branches receive the image term.** The reference branches must keep reproducing the source image, so they are left unchanged.
- **The image term acts only on the object.** It is multiplied by the target mask at the token resolution of each layer, so it cannot alter the background.
- **The adapter sees only the object.** Its input is the bounding box of the source object enlarged by ten per cent, with every pixel outside the object mask set to mid-grey, so that the embedding describes the object rather than its surroundings.

One property of the released configuration decides how strongly the adapter acts. Guidance normally amplifies a conditioning signal by the scale w, but in FreeFine guidance acts only in the region of

Table 10.1: Identity conditioning with IP-Adapter at four scales λ; $\lambda = 0$ is the baseline.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
λ	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
0 (baseline)	0.8523	0.0416	14.01	0.8391	0.0585	19.47
0.5	0.8523	0.0416	14.68	0.8396	0.0584	21.49
0.8	0.8518	0.0416	14.86	0.8394	0.0584	18.75
1.1	0.8512	0.0417	15.47	0.8393	0.0585	19.43
1.4	0.8502	0.0417	15.23	0.8390	0.0586	21.54

	Translations ($n = 67$)		Whole subset ($n = 200$)					
λ	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
0 (baseline)	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1278
0.5	0.9590	0.0495	0.9154	0.9656	0.0488	132.08	1637.46	0.1321
0.8	0.9589	0.0496	0.9153	0.9656	0.0489	132.11	1637.46	0.1303
1.1	0.9587	0.0497	0.9150	0.9656	0.0490	132.15	1637.56	0.1279
1.4	0.9585	0.0498	0.9147	0.9655	0.0490	132.27	1637.74	0.1286

Background consistency on the hard groups: 0.9664–0.9669 (rotations) and 0.9624–0.9631 (resizings). Kernel distance is not seeded.

Eq. (5.9), which excludes the object; inside the target mask the prediction is the unconditional one. Because the image term is added to both edit branches, the unconditional one included, it acts on the object at its nominal scale λ. In the usual use of the adapter the unconditional branch receives an empty image and the image term is amplified by w; here the scales studied are the actual strengths of the conditioning. Four scales were tested, $\lambda \in \{0.5, 0.8, 1.1, 1.4\}$, which cover the range in which image prompts are normally used and go beyond it.

10.1.2 Results

The adapter is active: its images differ from the baseline by 0.1 to 0.9 grey levels per pixel on average, increasingly with the scale. Table 10.1 reports the four scales.

No scale raises subject consistency on any group by more than 0.0005, one tenth of the threshold. On hard resizings every scale stays within 0.0005 of the baseline. On hard rotations subject consistency falls steadily as the scale grows, from 0.8523 at $\lambda = 0.5$ to 0.8518, 0.8512 and 0.8502, and on translations and the whole subset it falls slightly as well. Mean distance on hard resizings reads 21.49, 18.75, 19.43 and 21.54 against 19.47, without order; on hard rotations every scale raises it, by 0.67 to 1.46 pixels. The background, the warping error and the realism metrics hardly move.

10.1.3 Outcome

The refinement does not lack the object's identity. On translations, where no correspondence problem arises, the refinement already reproduces the object with a subject consistency of 0.959; an explicit description of the object has nothing to add to that, and it adds nothing here. Hypothesis 2 is rejected.

Why stronger identity lowers consistency on rotations. A pooled CLIP embedding summarises an object as a whole (its category, colours, materials and style), and the adapter was trained to generate images that match such a summary, in the views in which objects are usually photographed. For a resized object this pull agrees with the composite, since the object keeps its orientation. For a rotated object it competes with the rotated pose that the composite and TCA impose, and the more strongly the adapter is weighted, the further the output drifts from the specific, rotated source. The loss appears on rotations and not on resizings, as this explanation predicts. The spread of mean distance on hard resizings, almost three pixels without order, is the variation of the metric described in Experiment 3.

A higher-capacity adapter. The adapter summarises the object in four tokens derived from one pooled vector, which has no spatial layout. A variant released with it, IP-Adapter Plus, replaces the linear projection by a perceiver resampler [54] that reads the 257 patch tokens of the encoder's penultimate layer and produces sixteen tokens that keep spatial structure. It was implemented and verified to load the released checkpoint with no missing or unexpected parameters, but not evaluated: the base adapter had already shown that strengthening the identity signal lowers consistency, and the Plus variant supplies the same kind of information at a finer grain.

Next. Identity is not missing. The same channel can, however, carry more than identity if what the adapter is shown depends on the edit.

10.2 Experiment 9: Geometry through image tokens

10.2.1 Change

Since the affine map of every edit is known, the adapter can be shown the object exactly as it should look after the edit. The adapter was run at $\lambda = 0.6$, in the low range of Experiment 8 where it left every group within 0.0005 of the baseline, with three inputs.

- **Source crop**: the input of Experiment 8, as a control.
- **Warped crop**: the same crop warped with the affine map of the edit, so that the adapter sees the object at its target pose.
- **Crop with line drawing**: the four tokens of the source crop, which describe the identity of the object, concatenated with four tokens computed from a line drawing of the warped crop, which describe its outline at the target pose, multiplied by a weight $g \in \{0.5, 1.0\}$. CLIP describes sketches and line drawings meaningfully [10], and a drawing isolates shape from colour and texture.

The line drawing is the Sobel gradient magnitude of the warped crop, normalised and inverted so that edges are dark on white. Gradient magnitude is the first stage of a histogram of oriented gradients (HOG) [55]; the orientation histograms that complete a HOG descriptor are not computed, so the drawing is a plain edge-strength map.

Table 10.2: Geometry through image tokens: IP-Adapter at $\lambda = 0.6$ with three inputs.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Baseline	0.8523	0.0416	14.01	0.8391	0.0585	19.47
Source crop	0.8521	0.0416	14.72	0.8396	0.0584	18.71
Warped crop	0.8524	0.0416	14.67	0.8395	0.0584	19.62
Line drawing, $g = 0.5$	0.8524	0.0415	14.82	0.8396	0.0584	20.73
Line drawing, $g = 1.0$	0.8523	0.0415	14.70	0.8395	0.0584	19.86

	Translations ($n = 67$)		Whole subset ($n = 200$)					
	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
Baseline	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1283
Source crop	0.9590	0.0496	0.9154	0.9656	0.0488	132.09	1637.49	0.1304
Warped crop	0.9590	0.0496	0.9154	0.9656	0.0488	132.07	1637.60	0.1313
Line drawing, $g = 0.5$	0.9590	0.0495	0.9154	0.9655	0.0488	132.09	1637.22	0.1274
Line drawing, $g = 1.0$	0.9590	0.0495	0.9154	0.9656	0.0487	132.05	1637.45	0.1323

Background consistency of the four variants stays within 0.0008 of the baseline on every group. Kernel distance is not seeded.

10.2.2 Results

Table 10.2 shows no effect. Subject consistency on the whole subset is 0.9154 in all five rows, every variant stays within 0.0005 of the baseline on the hard groups, and the images change by less than half a grey level per pixel. The warped crop and the source crop give indistinguishable results on every metric, and adding the line drawing changes nothing. Mean distance on hard resizings spans 18.71 to 20.73 across the four variants, on either side of the baseline.

10.2.3 Outcome

The warped crop shows the adapter exactly the geometry that the edit requires, and the output does not change. The reason lies in the encoder. CLIP is trained contrastively to match images with their captions, and a caption names an object and its attributes but rarely its orientation or its size in the frame. The embedding is therefore largely invariant to exactly the properties that a geometric edit changes: a rotated or rescaled copy of the object is encoded into nearly the same four tokens. The same holds for the line drawing, which tells the encoder that it sees an outline of the object, not where the parts of that outline lie. The image-token channel cannot carry the geometry of the edit, whatever it is shown; geometry has to be supplied through a channel that is aligned with the image, which is what ControlNet provides.

10.3 Experiment 10: Geometry through residual feature maps

10.3.1 Change

ControlNet. ControlNet [36] conditions a frozen diffusion model on a spatial map such as an edge map, a depth map or a body pose. It keeps the pretrained U-Net frozen and trains a copy of its encoder, which receives the noisy latent together with the conditioning map, first reduced to latent resolution by a small convolutional network. The copy is connected to the frozen network through *zero convolutions*, 1×1 convolutions initialised to zero, so that training starts from the unmodified model. At inference its outputs are added to the twelve skip connections and to the middle block of the frozen U-Net,

$$h_\ell \leftarrow h_\ell + s\, r_\ell(z_t, t, c, e), \tag{10.2}$$

where h_ℓ is the feature map at connection ℓ, r_ℓ the corresponding output of the copy, e the conditioning map and s the conditioning scale. Because the residuals are aligned with the image, they can impose the outline and internal structure of the object at an exact position and pose, which is precisely what the image tokens could not do.

Conditioning map and integration. The coarse composite already shows the object at its exact target pose, so its edges describe the geometry that the refinement should respect. The map e is the Canny edge map [56] of the composite, with thresholds 100 and 200, restricted to the target mask dilated by a 25×25 square; the restriction keeps the edges of the background, which FreeFine handles itself, from competing with it. The released edge-conditioned ControlNet for SD-1.5 is used with $s = 0.9$. The residuals are computed for the two edit branches only (the reference branches must keep reproducing the source) and are not applied during inversion, which must reconstruct its inputs faithfully. To fit in the memory of the GPU, the ControlNet weights are held in half precision and their residuals converted to single precision before they are added; the rest of the pipeline stays in single precision.

Three variants. ControlNet was run alone, which supplies structure only; with the adapter of Experiment 8 at $\lambda = 0.6$, so that the edge map fixes the structure and the image prompt the appearance; and with the adapter and the object label as the prompt at $w = 10$, which runs ControlNet in the regime it was trained for, always with a text prompt. The component responsible for an effect is identified by comparing the combinations with ControlNet alone and with the single components measured earlier.

10.3.2 Results

ControlNet is by far the most active channel so far: it changes the images by 1.3 to 3.4 grey levels per pixel on average (Table 10.3). It lowers subject consistency on hard rotations by 0.0058 to 0.0061, more than the threshold, and on translations by 0.0041 to 0.0047. It raises the warping error on every group, from 0.0416 to 0.0441 on hard rotations and from 0.0488 to about 0.0512 on the whole subset, and it worsens FID by 0.9 to 1.0 and the DINOv2 distance by 1.9 to 2.5. On hard resizings subject consistency rises by 0.0011 to 0.0014, well below the threshold. The one favourable number, a mean distance of 17.91 on hard resizings, belongs to the variant that also uses the object label; guidance alone gave 17.75 (Experiment 5) and ControlNet alone gives 19.28.

Table 10.3: Geometry through residual feature maps: an edge-conditioned ControlNet at scale $s = 0.9$, alone and in combination.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Baseline	0.8523	0.0416	14.01	0.8391	0.0585	19.47
ControlNet	0.8465	0.0441	14.07	0.8403	0.0602	19.28
ControlNet + adapter	0.8465	0.0441	14.31	0.8405	0.0603	19.19
ControlNet + adapter + label, $w = 10$	0.8462	0.0442	14.14	0.8402	0.0605	17.91

	Translations ($n = 67$)		Whole subset ($n = 200$)					
	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
Baseline	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1237
ControlNet	0.9549	0.0521	0.9120	0.9659	0.0511	133.31	1639.06	0.1370
+ adapter	0.9547	0.0521	0.9119	0.9659	0.0512	133.23	1639.33	0.1447
+ adapter + label	0.9543	0.0523	0.9116	0.9658	0.0513	133.15	1638.71	0.1401

Background consistency on the hard groups: 0.9686–0.9689 (rotations, baseline 0.9664) and 0.9624–0.9627 (resizings, baseline 0.9631). Kernel distance is not seeded.

10.3.3 Outcome

Structure imposed from outside competes with the transplant. Two sources of structure act on the object at once. ControlNet imposes the edges of the object at the target pose from the first refined step. TCA, whose weight is one at that step (Eq. (5.6)), transplants the features of the source object, and those features describe it at its source pose: the features of Stable Diffusion are not equivariant to rotation, so a rotated object is not described by rotated features. The edge map adds a disagreement of its own, because the edges of the composite include the artefacts of resampling: jagged outlines, softened internal edges and the seam around the inserted object. The network settles on a compromise that matches neither source: further from the composite, so the warping error rises, and further from the source object, so subject consistency falls. The losses are largest on rotations, where transplanted features and imposed edges differ in orientation, and absent on resizings, where they differ only in scale. The only gain in mean distance comes from guidance, and ControlNet adds nothing to it.

Next. The remaining entry point is the transplant itself, where the geometry of the edit should matter most.

10.4 Experiment 11: Geometry inside the attention transplant

10.4.1 Change

How TCA retrieves the source object. In the object term of TCA, Eq. (5.4), each edit query $\mathbf{q}_p$ attends to the keys of the source object and receives a weighted average of their values. Two properties of this computation decide what geometry can do inside it. *Retrieval is by content*: the weights depend only on feature similarity, since the attention of SD-1.5 has no positional term, and rearranging

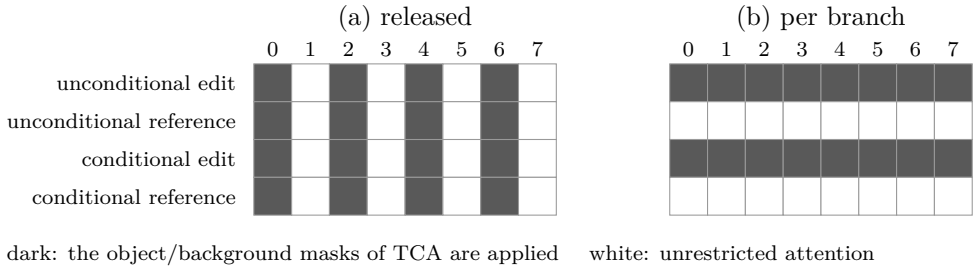


Figure 10.2: Which heads of which branch receive TCA's object and background masks (columns are the eight heads of a layer). (a) Released: the per-branch masks are tiled over heads while the rows are ordered by branch, so the even heads of every branch are masked and the odd heads are not. (b) The layout implied by the per-branch construction and Eq. (5.5): every head of the edit branches is masked and the reference branches are unrestricted.

the source tokens together with their mask entries would leave the output unchanged (Eq. (3.17)). *Retrieval aggregates*: every query receives an average over the whole source object, dominated by the tokens most similar to it. Both properties make TCA blind to the pose of the edit: nothing ensures that the handle of a rotated cup draws on the handle of the source cup rather than on another part that looks similar. That blindness is what the four mechanisms below address.

How TCA's masks reach the attention heads. Every mechanism below changes TCA's masks or its logits, so the path by which the masks reach the attention computation had to be traced exactly, and tracing it revealed a property of the released implementation. Each attention layer of SD-1.5 splits its features into $H = 8$ heads. The four branches of a refinement step are processed together (§5.4.1), and the implementation stacks branches and heads into one dimension of $4H = 32$ rows, ordered branch by branch: rows 0–7 are the heads of the first branch, rows 8–15 those of the second, and so on. The masks are built per branch (the object mask and its complement for the two edit branches, no restriction for the two reference branches) and then expanded to 32 rows by tiling the stack of four masks eight times. Row k therefore receives the mask built for branch $k \bmod 4$ while it belongs to branch $\lfloor k/H \rfloor$. For head h of branch b,

$$(bH + h) \bmod 4 \;=\; h \bmod 4 \qquad (H = 8), \tag{10.3}$$

independently of the branch: heads 0, 2, 4 and 6 of *every* branch receive the edit-branch masks, and heads 1, 3, 5 and 7 of every branch receive none (Figure 10.2). In the edit branches only half of the heads separate object from background as Eq. (5.5) prescribes; the other half attend to the whole reference image. In the reference branches, which should reproduce the source image's own attention, half of the heads are restricted as though they were edit queries. Running the released attention code, unmodified, on random inputs of the shape of a 32×32 decoder layer confirms the analysis (Table 10.4).

The reproduction of Chapter 7 ran the released code, so its results, which match the published ones, were produced with this layout. The mechanisms of this experiment are built on the layout that the per-branch construction implies, Figure 10.2(b), so that they act exactly where TCA is meant to act, and the per-branch layout is also run on its own, which measures the effect of the property itself.

Four places to add the geometry. The affine map M of every edit is known exactly, and the attention path of TCA offers four points at which it can be used (Figure 10.3). It can change the image

Table 10.4: Heads receiving the object mask, measured on the released attention code. A head counts as masked when none of its attention falls outside the source object.

	Released	Per-branch layout
Edit-branch heads masked	8 of 16	16 of 16
Reference-branch heads masked	8 of 16	0 of 16
Reference output vs. its own self-attention, max. $\lvert\Delta\rvert$	0.72	3×10^{-8}

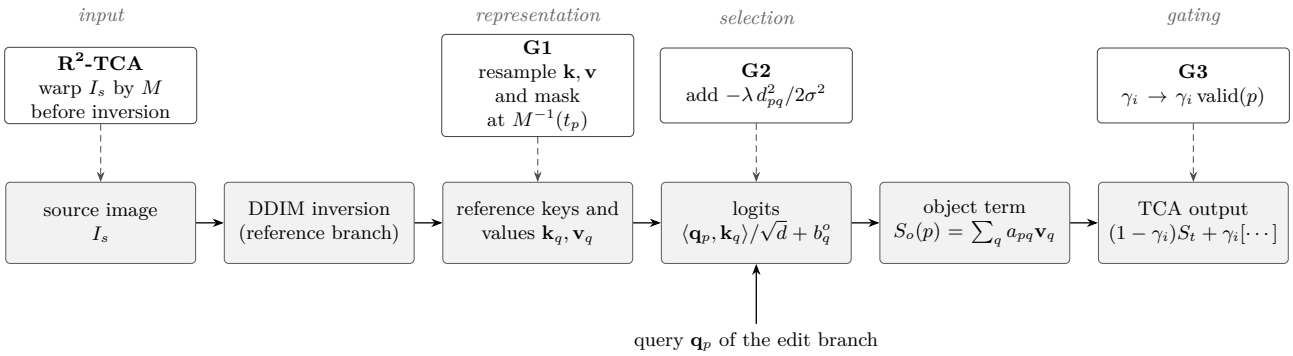


Figure 10.3: The attention path of TCA's object term (lower row) and the point at which each geometry-aware mechanism acts on it (upper row). R^2-TCA changes the image the reference is computed from, G1 the reference features, G2 the logits that select among them, and G3 the weight of the transplant in each token; $d_{pq} = \|t_p - M(s_q)\|$.

from which the reference branch is computed (the *input*), the reference features that the queries read (the *representation*), the logits that decide which of those features a query reads (the *selection*), or the weight with which the transplant enters each token (the *gating*). One mechanism was designed for each point. All four act in the six blocks that TCA modulates, at 32×32 and 64×64, in every refined step, and the TCA weight γ_i of Eq. (5.6) still applies on top of them.

They share one notation. Write t_p for the centre of target token p and s_q for the centre of source token q, in the pixel coordinates of the 512×512 image and with the pixel convention of the warp that builds the composite. Distances are measured in *token cells*, the spacing of the token grid: 8 pixels at 64×64 and 16 pixels at 32×32. The affine map M sends source coordinates to target coordinates, so $M(s_q)$ is where the edit moves the content of source token q, and $M^{-1}(t_p)$ is the source position whose content belongs at target token p. These positions are computed once per edit and per resolution.

G2: a correspondence bias on the logits

The most direct way to tell TCA which source token belongs to a query is to favour that token in the softmax. G2 adds to every logit of the object attention a penalty that grows with the distance between the query's position and the position to which M sends the key's source token:

$$\ell_{pq} = \frac{\langle \mathbf{q}_p, \mathbf{k}_q \rangle}{\sqrt{d}} + b_q^o - \lambda \, \frac{\|t_p - M(s_q)\|^2}{2\sigma^2}, \tag{10.4}$$

with $\sigma = 1.5$ cells and $\lambda \in \{4, 8\}$. The penalty is zero for a source token that lands exactly on the query and grows with the square of the distance (Figure 10.4). Relative to a token of equal

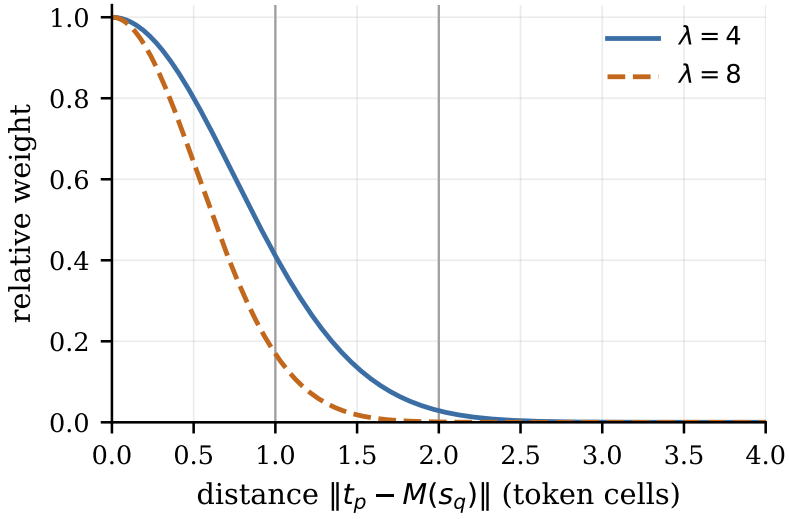


Figure 10.4: The factor by which the correspondence bias of Eq. (10.4) multiplies the softmax weight of a source token, as a function of the distance between the query and the position to which M sends the token ($\sigma = 1.5$ cells). One cell is 8 pixels at 64×64 and 16 pixels at 32×32.

similarity that lands on the query, a token that lands one cell away keeps 0.41 of its weight at $\lambda = 4$ and 0.17 at $\lambda = 8$, and a token two cells away keeps 0.03 and 0.0008. Feature similarity still acts inside that neighbourhood, so a query chooses among the source tokens around its geometric counterpart instead of among all the tokens of the object. The bias is applied to the two edit branches only.

The strength λ moves G2 between two extremes. At $\lambda = 0$ it is TCA: retrieval by content over the whole object. As $\lambda \to \infty$ the softmax concentrates on the source token nearest to the query's pre-image, $q^\star(p) = \arg\min_q \|t_p - M(s_q)\|$, and $S_o(p) \to \mathbf{v}_{q^\star(p)}$: every query copies the value of its geometric counterpart. That limit is a nearest-neighbour warp of the source features, that is, the coarse composite carried out in feature space. G2 is the only one of the four mechanisms that makes retrieval depend on position, and the two strengths test how far along this range the transplant should be moved.

G1: warped keys and values

G2 changes which source tokens a query prefers; G1 changes the source tokens themselves. The reference branch computes its keys and values at the source pose, so the feature that describes a part of the object sits where that part was before the edit. G1 moves every feature to where the edit puts it. For each target token p it reads the key and value at the source position $M^{-1}(t_p)$ by bilinear interpolation of the reference feature map, which is zero outside the frame,

$$\tilde{\mathbf{k}}_p = \mathbf{k}(M^{-1}(t_p)), \qquad \tilde{\mathbf{v}}_p = \mathbf{v}(M^{-1}(t_p)), \tag{10.5}$$

and it warps the source object mask in the same way with nearest-neighbour sampling, so that token p is an admissible key exactly when $M^{-1}(t_p)$ lies on the source object. The object term of the edit branches then reads the warped field, $S_o^{\mathrm{G1}}(p) = \sum_{p'} \mathrm{softmax}_{p'}\big(\langle \mathbf{q}_p, \tilde{\mathbf{k}}_{p'} \rangle / \sqrt{d} + \tilde{b}^o_{p'}\big)\, \tilde{\mathbf{v}}_{p'}$; the background term is unchanged.

Because retrieval is by content, moving keys and values together does not change *how* a query chooses:

by Eq. (3.17), a pure rearrangement of the source tokens would leave TCA's output exactly as it was. What G1 changes is *what there is to choose from*. Its features are interpolated between neighbouring source tokens; they are repeated where the object is enlarged, since an object scaled by s covers s^2 times as many tokens; and they are thinned where it is reduced, since the source tokens between the sample points are skipped. Even the translations of GeoBench, which are whole pixels but rarely whole token cells, are interpolated. G1 therefore tests whether TCA benefits from reference features that have already been brought to the target pose.

G3: a validity gate

Some target tokens have no counterpart on the source object: a part of the target region that the warped object does not reach, the band around it, and the background beyond. G1 still gives these tokens a transplant, read from wherever the warp happens to point. G3 withholds the transplant from them. A target token is valid if some token of the source object is mapped to within $r = 2$ cells of it, and for every other token the TCA weight is set to zero:

$$\gamma_i(p) = \gamma_i \cdot \mathrm{valid}(p), \qquad \mathrm{valid}(p) = \mathbb{1}\Big[\min_{q:\, b_q^o=0} \|t_p - M(s_q)\| < r\Big]. \tag{10.6}$$

An invalid token is generated by the edit branch's own self-attention, as in the late steps of the schedule. The two-cell tolerance keeps valid the tokens along the object's outline, where the discretisation of the token grid shifts the mapped positions by up to about a cell. Because the gate multiplies the whole TCA output of a token, it also switches off the background term everywhere beyond two cells from the object. It is used together with G1, so that the warped transplant acts only where the warp has a source.

R^2-TCA: a pose-aligned reference

G1 moves features after the network has computed them at the source pose. R^2-TCA moves the image instead, so that the network computes the features at the target pose. The source image is warped by M with the same bilinear warp that builds the composite, and this warped image $\mathcal{W}_A(I_s)$, not the source, is inverted and reconstructed by the reference branch; the target mask replaces the source mask as the reference's object mask. TCA itself is unchanged: its queries still retrieve by content, now from a reference in which the object already stands where the edit places it.

One identity makes the mechanism precise. By Eq. (5.3), the coarse composite equals $\mathcal{W}_A(I_s)$ inside the target mask, so inside M_t the reference of R^2-TCA is, pixel for pixel, the object of the coarse composite. R^2-TCA therefore transplants into the object the network's own features of the composite's object, resampling artefacts included, instead of the features of the undistorted source. Outside the object the reference no longer matches the scene: the warp moves the whole source image, so the background term reads a displaced background, and where the warp uncovers the frame it reads black borders. R^2-TCA tests whether the failure of TCA comes from the mismatch of pose between its reference and the edit; if it does, removing the mismatch should help.

Table 10.5 summarises the four mechanisms.

Implementation and verification. The mechanisms are implemented in the released attention code, behind switches, on four principles.

Table 10.5: The four mechanisms that give TCA the geometry of the edit.

Mechanism	Level	What it changes	Retrieval	Parameters
G2, bias	selection	the logits of the object attention, Eq. (10.4)	content and position	$\lambda \in \{4, 8\}$, $\sigma = 1.5$
G1, warp	representation	reference keys, values and object mask, Eq. (10.5)	content	none
G3, gate	gating	the TCA weight of each token, Eq. (10.6)	unchanged	$r = 2$
R^2-TCA	input	the image of the reference branch and its object mask	content	none

1. **The per-branch mask layout** of Figure 10.2(b) is used, so that every head of the edit branches performs TCA's separation and the reference branches are unrestricted.

2. **Only the edit branches are modified.** The keys, values and masks of the reference branches are left exactly as they are, so that the features TCA draws on are the source image's own.

3. **R^2-TCA uses the target mask** as its reference object mask, since in the warped reference the object occupies the target region.

4. **Token positions follow the pixel convention** of the warp that builds the composite, so that M applied to token centres is exactly the map applied to the image.

Six variants were run: the per-branch layout alone, which is the reference point of the mechanisms, and the per-branch layout with G2 at $\lambda = 4$ and $\lambda = 8$, with G1, with G1 and G3, and with R^2-TCA. Each mechanism is thus compared with a pipeline that differs from it by the mechanism only. Before any image was generated, the implementation was tested against the released source code on the checks of Table 10.6; all of them pass. The study was scored with both stochastic metrics seeded, so every value of Table 10.7, kernel distance included, is reproducible and comparable across rows. The variants were generated in two groups of three, and the released baseline, scored with each group, reproduced its earlier values on every deterministic metric of every group and gave the same seeded kernel distance both times.

10.4.2 Results

Table 10.7 reports the study.

The per-branch layout on its own. Used without any mechanism, this layout changes almost nothing. Subject consistency moves by +0.0002 on hard rotations, 0 on hard resizings, +0.0007 on translations and +0.0002 on the whole subset; the warping error and background consistency move by at most 0.0004, and the DINOv2 distance improves by about one unit. Mean distance moves by +0.79 and −1.35 pixels on the two hard groups, within the variation of the metric.

The mechanisms. Relative to the per-branch layout, every mechanism lowers subject consistency on every group.

Table 10.6: Checks of the implementation against the released code, run before any image was generated.

Check	Result
All switches off: TCA output against the released code	bit-identical
Per-branch layout: heads masked; reference output against its own self-attention	Table 10.4
G2, G1, G3: reference-branch output against the per-branch layout	bit-identical
G2: position of the largest bias for each query	the source token mapped onto it
G1: token warp against the image warp that builds the composite	exact on every token (translation, 90° rotation)
G3: valid tokens	exactly those with an object token mapped within r
R^2-TCA: reference image and object mask	the warped source; the target mask
Affine map passed to TCA against the map that builds the composite	identical

- **The correspondence bias** does so most strongly, and more the stronger it is: by 0.0155 and 0.0189 on hard rotations, 0.0094 and 0.0111 on hard resizings, and 0.0152 and 0.0188 on translations, up to four times the threshold. It raises the warping error, from 0.0416 to 0.0467 and 0.0474 on hard rotations, and FID by 1.6 and 2.1. It lowers mean distance on hard resizings to 17.47 and 17.42, against 18.12, a reduction smaller than the variation of the metric and paid for with 0.009 to 0.011 of subject consistency.

- **The warped keys and values** cost 0.0063 on hard rotations, 0.0039 on hard resizings and 0.0025 on translations, and raise mean distance on hard resizings by 1.54 pixels. The validity gate adds little to the loss of consistency, raises mean distance on hard resizings by a further 3.75, and lowers background consistency on the whole subset from 0.9655 to 0.9618, since it switches off the transplant of the background away from the object.

- **The pose-aligned reference** is the mildest: 0.0021 on hard rotations, 0.0039 on hard resizings and 0.0003 on translations. It gives the lowest warping error on hard rotations, 0.0395, the best FID and DINOv2 distance of the table, 131.94 and 1634.19, and lowers mean distance on hard rotations by 0.32 pixels, within the variation of the metric. It lowers background consistency, most on hard resizings, from 0.9627 to 0.9556: the background term of TCA now draws on a displaced background with black borders.

Figure 10.5 places every mechanism relative to the per-branch layout on the two hard groups.

10.4.3 Outcome

The released head layout is inconsequential. TCA performs as well with its separation on half of the heads as on all of them. The reproduction, and every result obtained on the released layout, stands.

Table 10.7: Geometry inside the attention transplant. The mechanisms are built on the per-branch layout, which is therefore their reference point; the released baseline is given for comparison.

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Released baseline	0.8523	0.0416	14.01	0.8391	0.0585	19.47
Per-branch layout	0.8525	0.0416	14.80	0.8391	0.0584	18.12
+ G2, $\lambda = 4$	0.8370	0.0467	14.80	0.8297	0.0607	17.47
+ G2, $\lambda = 8$	0.8336	0.0474	15.23	0.8280	0.0609	17.42
+ G1	0.8462	0.0421	15.70	0.8352	0.0583	19.66
+ G1 + G3	0.8454	0.0417	15.19	0.8349	0.0580	23.42
+ R^2-TCA	0.8504	0.0395	14.47	0.8352	0.0584	20.12

	Translations ($n = 67$)		Whole subset ($n = 200$)					
	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
Released baseline	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1268
Per-branch layout	0.9597	0.0494	0.9156	0.9656	0.0487	132.37	1635.85	0.1263
+ G2, $\lambda = 4$	0.9445	0.0542	0.9043	0.9657	0.0527	134.00	1639.72	0.1352
+ G2, $\lambda = 8$	0.9409	0.0551	0.9017	0.9657	0.0533	134.45	1640.99	0.1362
+ G1	0.9572	0.0496	0.9119	0.9655	0.0490	132.57	1637.03	0.1252
+ G1 + G3	0.9571	0.0495	0.9116	0.9618	0.0488	132.78	1641.78	0.1343
+ R^2-TCA	0.9594	0.0490	0.9146	0.9635	0.0478	131.94	1634.19	0.1269

Kernel distance is seeded. Background consistency on the hard groups: 0.9636–0.9668 (rotations) and 0.9556–0.9630 (resizings); the lowest values in both groups belong to R^2-TCA.

The geometry does not help. Giving the attention transplant the exact geometry of the edit lowers subject consistency at every point of the attention path where it can enter. The two kinds of mechanism fail for different reasons, and the reasons explain what TCA is.

A pose-aligned reference gives TCA nothing it lacks. R^2-TCA gives the transplant the most faithful form of the geometry available: a reference in which the object already stands at its target pose, with features computed by the network itself. Inside the target mask that reference is the composite's own object, so the transplant copies the composite's appearance into the edit; this is why R^2-TCA has the lowest warping error of the table on hard rotations, 0.0395. Yet it scores slightly below its reference point in subject consistency on every group. G1, which brings the reference to the target pose in feature space, costs more, because its features are interpolated rather than computed. Removing the mismatch of pose between reference and edit therefore does not help: retrieval by content already finds the source features that match each query, wherever they lie, and replacing the undistorted source by its resampled copy only imports the artefacts of the resampling.

Imposing correspondence removes aggregation. The correspondence bias lowers consistency in proportion to its strength, and it lowers it on translations as much as on hard rotations, 0.0152 and 0.0188 against 0.0155 and 0.0189. For a translation the correspondence is exact (the checks of Table 10.6 confirm that the bias peaks on the right source token), so the loss cannot come from wrong geometry. It comes from the restriction itself. At $\lambda = 4$ a query draws in effect only on the source tokens within about two cells of its counterpart instead of on the whole object (Figure 10.4); the average that TCA forms over many similar tokens is replaced by a local copy of a few, a noisier and

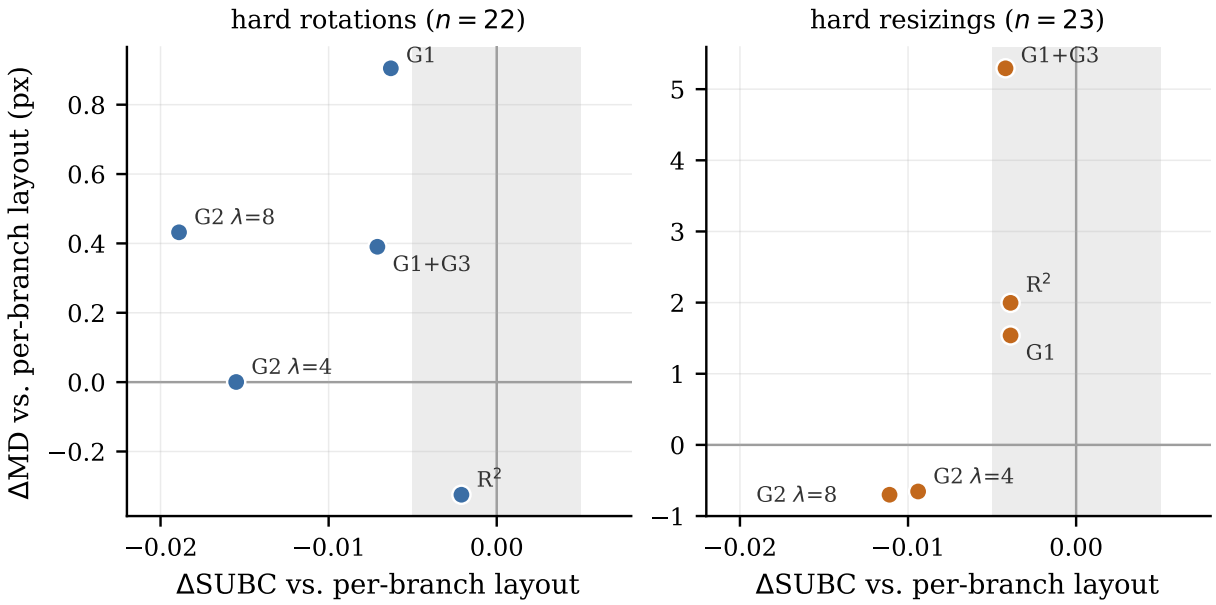


Figure 10.5: Each geometry-aware mechanism relative to the per-branch layout on the two hard groups: change in subject consistency against change in mean distance, where lower is better. The shaded band marks changes of subject consistency within ±0.005.

less complete description of the object. The larger λ, the closer G2 comes to its limit, a nearest-neighbour copy of the source features, and the larger the loss. The output drifts away from the source object, so subject consistency falls; away from the composite, so the warping error rises; and away from natural images, so FID rises.

Conclusion. TCA does not match the parts of the target to the parts of the source, and it does not need to: its aggregation by content is what preserves the appearance of the object, independently of the pose. Giving it the geometry of the edit either changes nothing, when the geometry only moves the features it reads, or harms it, when the geometry restricts what it may read. Hypothesis 3 is rejected through the attention transplant, in an implementation verified against the released code.

10.5 The entry points as a whole

Table 10.8 collects, for every entry point of this chapter and for the text prompt of Experiment 5, where the information acts and how far it moved subject consistency on the two hard groups.

The pattern is consistent. The only channel that produced a clear improvement on a hard group (the text prompt, which lowered mean distance on hard resizings by 1.73 pixels, more the stronger the guidance) is the only one that does not act on the object directly. The channels that act on the object either leave it unchanged, as the image tokens do, or move it away from its source, as the feature maps and the geometry-aware attention do, and they do so more strongly the more directly they constrain its structure.

Hypotheses 2 and 3 are rejected for every mechanism tested, at each of the three entry points that a

Table 10.8: Where information supplied from outside acts in the refinement, and the range of the change in subject consistency that its variants produced on the hard groups.

			ΔSUBC	
Entry point	Information	Where it acts	Hard rotations	Hard resizings
Text prompt with guidance (E5)	object label	a band around the object, not the object	+0.0005 to +0.0006	−0.0001 to 0.0000
(A) tokens, identity (E8)	object crop	the object, both edit branches	−0.0021 to 0.0000	−0.0001 to +0.0005
(A) tokens, geometry (E9)	warped crop, line drawing	the object, both edit branches	0.0000 to +0.0001	+0.0004 to +0.0005
(B) feature maps (E10)	edges at the target pose	all features of the edit branches	−0.0061 to −0.0058	+0.0011 to +0.0014
(C) self-attention (E11)	the affine map	the transplant of the object	−0.0189 to −0.0021	−0.0111 to −0.0039

Changes relative to the released baseline, except for self-attention, where they are relative to the per-branch layout. The threshold for an improvement is +0.005.

frozen U-Net offers. The refinement reproduces the identity of the object, and the exact geometry of the edit is already present in its input: the coarse composite shows the object at its exact target pose. The remaining explanation is that information present in the composite is lost during refinement. Chapter 11 tests it by anchoring the refinement to the composite.

Chapter 11

Hypothesis 4: Anchoring the Refinement to the Coarse Composite

Chapters 9 and 10 tested two explanations of the failure on hard rotations and resizings and found support for neither: no tested setting of the pipeline, and no tested way of supplying the identity of the object or the geometry of the edit, improves the hard groups. One explanation remains. The coarse composite already shows the object at its exact target pose, copied pixel for pixel from the source (§5.3); if refinement degrades what the composite gets right, keeping more of the composite should help.

> **Hypothesis 4.** Information present in the coarse composite is lost during refinement, and keeping the output closer to the composite improves the edit without a loss of realism.

Six experiments test it. Experiment 12 anchors the refinement to the composite while it denoises. Experiment 13 scores the composite itself, which measures how much there is to keep. Experiment 14 builds the first models that restore the composite's interior after refinement, chosen by edit type, and checks them on the held-out subset. Experiments 15 to 17 then establish how wide the refined boundary should be, what makes interior restoration work, and for which edit types it works.

11.1 Experiment 12: Anchoring during denoising

11.1.1 Change

The inversion of §5.4.1 computes, and stores, the latent of the coarse composite at every timestep of the refined window: a trajectory z_t^c that the sampler would retrace if nothing were modified. Latent preservation pulls the edit latent towards that trajectory after every refined step,

$$z_{t_{i+1}}^{\text{edit}} \leftarrow (1 - w\, m_{\text{var}}) \odot z_{t_{i+1}}^{\text{edit}} + w\, m_{\text{var}} \odot z_{t_{i+1}}^{c}, \qquad i = s, \ldots, 49, \tag{11.1}$$

with a strength $w \in \{0.3, 0.5, 0.7, 0.9\}$; $w = 0$ is the baseline. The blend acts on the edit latent only, and in the region where refinement departs from its input: m_{var} of Eq. (5.9), the band around the transformed object together with the area it vacated, where Local Perturbation injects noise and guidance acts. The object itself is refined deterministically and is not blended.

Because the blend is applied after every one of the fifteen refined steps, its effect compounds. Whatever the region gains from one denoising step, a fraction w of it is replaced by the composite's trajectory

Table 11.1: Latent preservation towards the composite's inversion trajectory, Eq. (11.1).

	Hard rotations ($n = 22$)			Hard resizings ($n = 23$)		
w	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
0 (baseline)	0.8523	0.0416	14.01	0.8391	0.0585	19.47
0.3	0.8536	0.0397	14.63	0.8409	0.0538	21.74
0.5	0.8535	0.0396	14.69	0.8404	0.0537	20.15
0.7	0.8535	0.0395	14.77	0.8400	0.0537	22.04
0.9	0.8535	0.0395	14.87	0.8399	0.0537	24.19

	Translations ($n = 67$)		Whole subset ($n = 200$)					
w	SUBC	WRAP_E	SUBC	BGC	WRAP_E	FID	FID_{dino}	KD
0 (baseline)	0.9590	0.0495	0.9154	0.9657	0.0488	132.28	1636.82	0.1228
0.3	0.9610	0.0468	0.9175	0.9631	0.0454	133.14	1641.78	0.1391
0.5	0.9612	0.0467	0.9174	0.9627	0.0453	133.38	1643.64	0.1462
0.7	0.9614	0.0467	0.9174	0.9625	0.0453	133.45	1644.17	0.1449
0.9	0.9614	0.0467	0.9174	0.9626	0.0453	133.51	1644.47	0.1420

Background consistency on the hard groups: 0.9661–0.9663 (rotations, baseline 0.9664) and 0.9593–0.9607 (resizings, baseline 0.9631). Kernel distance is not seeded.

at the next, so the region cannot drift far from that trajectory for any of the strengths tested. The four strengths are therefore not four independent doses.

11.1.2 Results

The mechanism is active: at $w = 0.3$ its images differ from the baseline by 1.1 to 4.7 grey levels per pixel on the first edits compared, and at $w = 0.9$ by 1.3 to 5.3. Table 11.1 reports the four strengths.

Latent preservation is the first change in this thesis that raises subject consistency on every group: by 0.0013 on hard rotations, 0.0018 on hard resizings, 0.0020 on translations and 0.0021 on the whole subset at $w = 0.3$. It lowers the warping error everywhere, by 5 to 8%. The gains do not grow with the strength: subject consistency is the same to within 0.0010 for every w. The costs are clear. FID rises by 0.9 to 1.2, the DINOv2 distance by 5.0 to 7.7 and the kernel distance by 0.016 to 0.023; background consistency on the whole subset falls by 0.0026 to 0.0032; and mean distance worsens on both hard groups, by 0.6 to 0.9 pixels on rotations and by 0.7 to 4.7 on resizings.

11.1.3 Outcome

A small gain that does not scale. Holding the object's surroundings at the composite nudges the object towards it as well (the warping error, measured inside the object, falls), and subject consistency follows, by about 0.002. The flat dose response has a simple cause: pulled back at every step, the region stays close to the composite's trajectory for any $w \geq 0.3$, which is also why the images change little more at $w = 0.9$ than at $w = 0.3$. The price is paid in realism, because the band around the object is exactly where refinement blends the inserted object into the scene.

Table 11.2: The refined baseline against the coarse composite scored as an output, on the development and held-out subsets.

	Development subset				Held-out subset			
	SUBC↑		MD↓		SUBC↑		MD↓	
Group	refined	composite	refined	composite	refined	composite	refined	composite
Translations	0.9590	0.9712	3.41	4.47	0.9595	0.9697	3.62	5.58
Rotations	0.8962	0.9056	9.63	11.17	0.9084	0.9160	8.44	8.98
Hard rotations	0.8523	0.8608	14.01	17.86	0.8633	0.8721	11.70	13.30
Resizings	0.8906	0.8967	10.93	10.27	0.8871	0.8935	13.91	14.43
Hard resizings	0.8391	0.8408	19.47	16.44	0.8024	0.8110	29.64	28.08
Whole subset	0.9154	0.9246	n/a	n/a	0.9182	0.9263	n/a	n/a

Group sizes: development 67, 66, 22, 67, 23, 200; held-out 66, 68, 24, 66, 22, 200.
Whole subset, refined against composite: background consistency 0.9657 against 0.9854 (development) and 0.9667 against 0.9859 (held-out); FID 132.28 against 132.67 and 132.15 against 132.96; DINOv2 distance 1636.82 against 1654.71 and 1624.90 against 1631.99; kernel distance (not seeded) 0.1283 against 0.1360 and 0.1117 against 0.1163.

What this experiment cannot tell. The blend never touches the object itself, and its strengths are not independent doses. The experiment therefore does not measure how much the edit would gain if the object were kept as it is in the composite. That quantity can be measured directly, by scoring the composite itself.

11.2 Experiment 13: The composite as an output

11.2.1 Change

No refinement at all: the coarse composite stored with the benchmark is scored with the same metric code as the output of a method, on both the development and the held-out subsets. Its warping error is zero by definition, since the composite is the reference of that metric. Comparing it with the refined baseline shows, group by group, what refinement adds to the composite and what it takes away.

11.2.2 Results

Table 11.2 and Figure 11.1 report the comparison. The composite scores a higher subject consistency than the refined output on every group of both subsets, by 0.0017 to 0.0122, and a higher background consistency, above 0.985 against about 0.966. It scores a worse realism: FID is 0.39 and 0.80 higher, and the DINOv2 distance 17.9 and 7.1 higher. On mean distance the groups divide. The composite is worse on translations and on rotations in both subsets, by 0.5 to 3.9 pixels, and better on hard resizings in both, by 3.0 and 1.6 pixels.

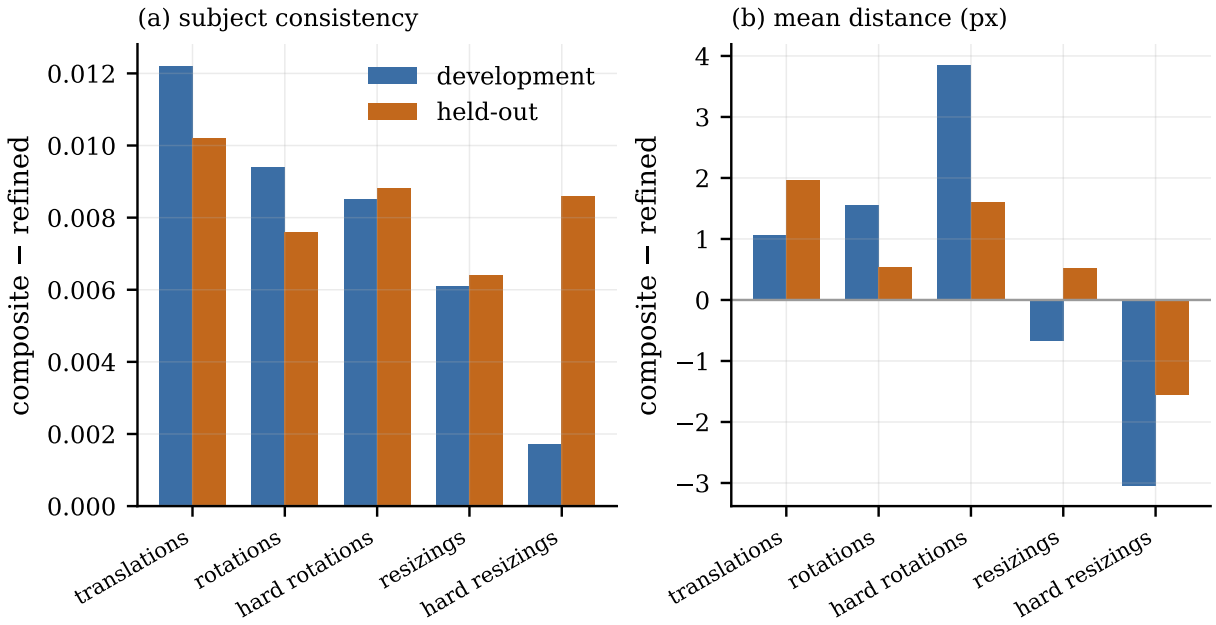


Figure 11.1: The coarse composite minus the refined baseline, per group and subset. (a) Subject consistency: the composite is higher everywhere. (b) Mean distance: the composite is worse on translations and rotations and better on hard resizings.

11.2.3 Outcome

The refined output is within one per cent of the composite on subject consistency. On hard rotations the baseline reaches 0.8523/0.8608 = 99.0% of the composite's subject consistency on the development subset, and 0.8633/0.8721 = 99.0% on the held-out subset; on the whole subset it reaches 99.0% and 99.1%. Subject consistency compares the edited object with the source object, and the composite contains the source pixels themselves, resampled to the target pose; the refined output falls short of that copy by about one per cent. This bounds what anchoring can achieve on the metric: the room left between refinement and simply copying the composite is 0.0085 and 0.0088 on hard rotations and 0.0017 and 0.0086 on hard resizings.

Refinement pays for realism with consistency. The composite is worse on every realism metric, because its object is inserted with a visible seam and carries the artefacts of resampling, and it is better on both consistency metrics, because it is the source copied into place. Background consistency is 0.02 higher for the composite because refinement regenerates the band around the object and decodes the whole image through the autoencoder, while the composite is neither regenerated nor decoded. Refinement converts part of the composite's consistency into realism; the question is whether some of the composite can be kept without paying the realism back.

Mean distance mixes geometry with appearance. On translations the composite is geometrically exact (every translation of GeoBench is a whole number of pixels, so the warp copies the object without resampling), and yet its mean distance is worse than that of the refined output, 4.47 against 3.41 and 5.58 against 3.62. The keypoint metric locates keypoints with diffusion features (§5.7), and it locates them more reliably in a refined, natural-looking image than in a composited one. A difference in mean distance between the composite and a refined image therefore mixes geometry with

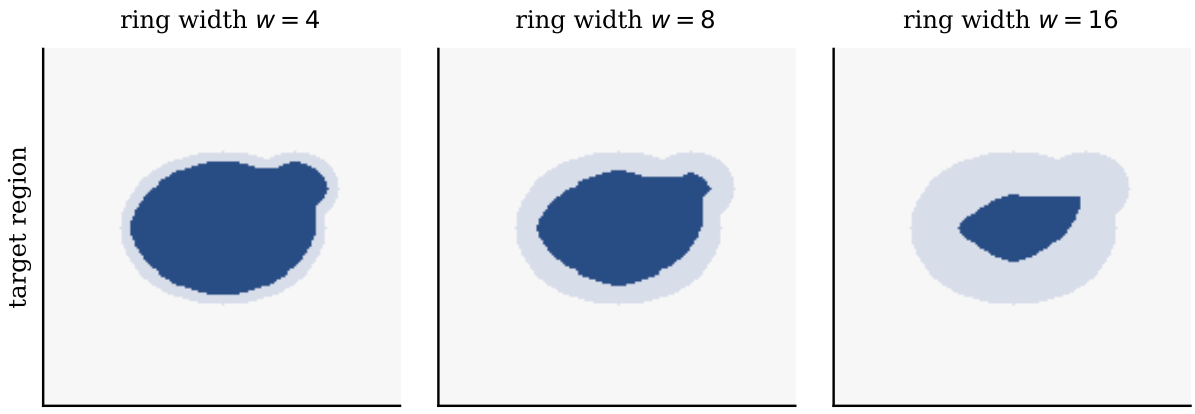


Figure 11.2: The construction of Eq. (11.2) for an object mask at three widths w: the dark interior is taken from the coarse composite, the light band of width w inside the outline, and everything outside the object, from the refined output.

appearance. With that caveat, one pattern holds on both subsets: on hard resizings the composite is better, so there refinement moves resized content away from where the transformation put it, as Experiment 5 found.

Next. The composite is worth keeping where it is reliable, that is, in the interior of the object, which it copies from the source, while refinement handles the boundary, where the composite's seam is visible. The next experiment builds that combination and chooses it by edit type.

11.3 Experiment 14: Routed models with interior restoration

11.3.1 Change

Interior restoration. After refinement, the interior of the object is replaced by the composite, and a band along the object's outline is left as refined:

$$\mathcal{R}_w(\hat{I}) = M_{\text{int}} \odot I_c + (1 - M_{\text{int}}) \odot \hat{I}, \qquad M_{\text{int}} = M_t \ominus B_{2w+1}, \tag{11.2}$$

where $\hat{I}$ is the refined output, $\ominus$ is binary erosion and B_{2w+1} a $(2w + 1) \times (2w + 1)$ square, so that the refined band is w pixels wide inside the outline (Figure 11.2). The composite supplies the object's interior, which it copies exactly from the source; the refined band blends the object into the scene, which the composite cannot do. Interior restoration needs no new generation: it is applied to images that refinement has already produced. An object narrower than $2w + 1$ pixels has no interior left after erosion, and its output is the refined image unchanged. This experiment uses $w = 12$.

The letter w now stands for three quantities, which never meet in one formula: the guidance scale (§3.7), the strength of latent preservation in Eq. (11.1), and, wherever interior restoration is concerned, the width of the refined band in pixels. Every value of w is given together with the quantity it belongs to.

Table 11.3: The first routed models on the development subset and, unchanged, on the held-out subset.

	Hard rotations		Hard resizings		Translations		Whole subset	
	SUBC↑	MD↓	SUBC↑	MD↓	SUBC↑	MD↓	SUBC↑	FID↓
Development subset								
Baseline	0.8523	14.01	0.8391	19.47	0.9590	3.41	0.9154	132.28
R1	0.8523	14.01	0.8391	17.75	0.9610	3.52	0.9161	132.50
COMP	0.8570	17.36	0.8372	18.35	0.9652	3.68	0.9196	132.47
MAX1	0.8523	14.01	0.8375	17.76	0.9674	3.71	0.9191	132.50
Held-out subset								
Baseline	0.8633	11.70	0.8024	29.64	0.9595	3.62	0.9182	132.15
R1	0.8633	11.70	0.8035	23.01	0.9617	3.68	0.9192	132.23
COMP	0.8668	13.62	0.8041	28.07	0.9639	3.94	0.9211	131.86
MAX1	0.8633	11.70	0.8050	23.50	0.9665	3.80	0.9212	131.49

Group sizes: development 22, 23, 67, 200; held-out 24, 22, 66, 200.
Whole subset: warping error 0.0488, 0.0480, 0.0249, 0.0308 (development) and 0.0455, 0.0445, 0.0227, 0.0290 (held-out) for the four rows; DINOv2 distance 1636.82, 1637.15, 1637.87, 1636.52 and 1624.90, 1626.22, 1625.28, 1624.89. Kernel distance is not seeded.

Routing by edit type. Experiments 5, 12 and 13 point to different treatments for different edits: guidance with the object label helps resizings, latent preservation helps translations, and on rotations the composite's mean distance is worse than refinement's. The type of an edit is read from its parameters (a rotation if $r_z \neq 0$, a resizing if $(s_x, s_y) \neq (1, 1)$, otherwise a translation), so a model can act differently for each type without any other information. Three models were built.

- **R1** routes the refinement itself: translations receive latent preservation at $w = 0.3$ (Experiment 12), resizings guidance with the object label at $w = 10$ (Experiment 5), and rotations the released pipeline.

- **COMP** applies interior restoration, Eq. (11.2), to the baseline output of every edit, with the composite stored with the benchmark.

- **MAX1** applies interior restoration to the output of R1, with the composite that R1 itself refined, and exempts rotations, which keep the baseline output.

All three were evaluated on the development subset and, without any change, on the held-out subset.

11.3.2 Results

R1 changes the images of translations and resizings and leaves every rotation identical to the baseline, as routing requires. Table 11.3 reports the three models on both subsets.

MAX1 raises subject consistency on the whole subset by 0.0037 on the development subset and 0.0030 on the held-out subset, and on translations by 0.0084 and 0.0070. It lowers mean distance on hard resizings by 1.71 and 6.14 pixels, as R1 alone does (1.73 and 6.63), and leaves rotations exactly as

they were. Its FID moves by +0.23 on the development subset and by −0.66 on the held-out subset. On translations its mean distance is 0.30 and 0.18 pixels worse. COMP, which restores the interior of rotations too, raises subject consistency on hard rotations by 0.0047 and 0.0035 but worsens their mean distance by 3.35 and 1.92 pixels.

11.3.3 Outcome

The first improvement that holds on unseen edits. MAX1 is the first model of the thesis that improves on the baseline in the same direction on both subsets: higher subject consistency on the whole subset and on translations, lower mean distance on hard resizings, and rotations unchanged by construction. The held-out subset played no part in building it, so the improvement is not a product of fitting the development subset. FID moves in opposite directions on the two subsets, so realism is unchanged within the resolution of 200 edits.

What is gained by construction. Interior restoration copies the composite into the object, so the warping error falls mechanically, and part of the gain in subject consistency does too, since that metric compares the object with the source whose pixels the composite contains (rule 4 of §8.5). The gains that are not mechanical are the ones that matter: realism does not fall although a restored interior replaces refined content, and the refined band still harmonises the object with the scene.

Rotations must be exempt. COMP shows what happens without the exemption. Restoring a rotated object's interior from the composite raises subject consistency by less than the threshold and worsens mean distance on hard rotations by 3.35 and 1.92 pixels, several times the cost of interior restoration on translations, 0.30 and 0.18 pixels for MAX1, where the restored interior is an exact copy. The resampled, rotated interior is located much less accurately by the keypoint metric than the refined one, and restoring it undoes what refinement achieves on rotations.

Next. Two properties of MAX1 were fixed without evidence: the width of the refined band, $w = 12$, and the use of latent preservation for translations. The next experiments establish the width, test what makes interior restoration work, and determine for which edit types it helps.

11.4 Experiment 15: The width of the refined band

11.4.1 Change

Interior restoration, Eq. (11.2), was applied to the baseline output of the translations with four widths, $w \in \{4, 8, 16, 24\}$, using the composite stored with the benchmark. Translations are studied first because there the composite is an exact copy of the source, so the width alone decides how much of the object is exact source and how much is refined. A thin band keeps more of the composite; a wide band leaves more to refinement.

Table 11.4: Interior restoration on translations for four widths of the refined band. The other edit types are unchanged.

	Translations ($n = 67$)						Whole subset ($n = 200$)	
Width	SUBC↑	BGC↑	WRAP_E↓	MD↓	FID↓	KD↓	FID↓	KD↓
Baseline	0.9590	0.9639	0.0495	3.41	200.04	0.0259	132.28	0.1268
$w = 4$	0.9670	0.9644	0.0115	3.96	199.99	0.0213	132.24	0.1258
$w = 8$	0.9661	0.9643	0.0181	3.89	200.33	0.0231	132.33	0.1260
$w = 16$	0.9640	0.9643	0.0277	3.54	200.96	0.0213	132.47	0.1259
$w = 24$	0.9628	0.9642	0.0346	3.42	201.20	0.0231	132.55	0.1260

Kernel distance is seeded. The DINOv2 distance on the whole subset stays within 1637.09–1637.64 (baseline 1636.82).

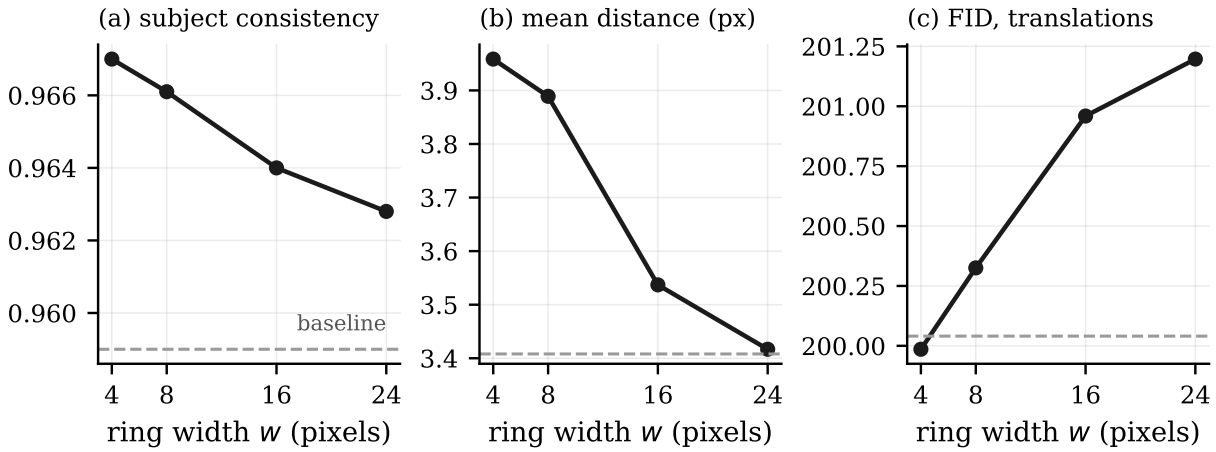


Figure 11.3: Interior restoration on translations as a function of the width w of the refined band; the dashed line is the baseline.

11.4.2 Results

The width trades subject consistency against mean distance (Table 11.4, Figure 11.3). At $w = 4$ subject consistency on translations rises by 0.0080, the warping error falls by 77%, and mean distance worsens by 0.55 pixels; as the band widens, the gain in subject consistency shrinks to 0.0038 at $w = 24$ and mean distance returns to the baseline value. FID on translations rises steadily with the width, from 199.99 at $w = 4$, the only width below the baseline, to 201.20 at $w = 24$; the whole subset follows the same order.

11.4.3 Outcome

The width decides how much of the object is exact source and how much is regenerated. The thinner the band, the more of the object is the composite's exact copy, which raises subject consistency and lowers the warping error by construction, and the more of the composite's keypoint behaviour appears in mean distance (Experiment 13). Realism does not suffer from a thin band: the narrowest band gives the best FID, so the part of the output that costs realism is the regenerated boundary, not the restored interior. For translations the thin band $w = 4$ is the best choice, with a declared cost of about half a pixel in mean distance. Whether interior restoration works because the composite is exact, and whether it can be applied to every edit type, are the questions of the next experiment.

Table 11.5: Scope: latent preservation and interior restoration applied to translations only or to every edit.

	Whole subset ($n = 200$)					Rotations ($n = 66$)		Resizings
	SUBC↑	MD↓	FID↓	FID_{dino}↓	KD↓	MD↓	FID↓	MD↓
Baseline	0.9154	7.60	132.28	1636.82	0.1268	9.63	205.96	10.93
Latent preservation, translations	0.9161	7.65	132.53	1636.51	0.1315	9.63	205.96	10.93
Latent preservation, all edits	0.9175	8.11	133.14	1641.78	0.1381	9.80	207.10	12.48
Restoration $w = 8$, translations	0.9177	7.56	132.33	1637.22	0.1260	9.63	205.96	10.93
Restoration $w = 8$, all edits	0.9207	8.68	132.46	1638.16	0.1264	11.30	206.55	10.90

Resizings: $n = 67$. Kernel distance is seeded. FID on resizings: 201.26 for the baseline, 201.99 with latent preservation on all edits and 200.92 with interior restoration on all edits.

Table 11.6: Warp quality: the translation composite built by the released bilinear warp, by an exact integer shift, or by the anti-aliased warp, followed by interior restoration with $w = 8$.

	Translations ($n = 67$)					
Composite	SUBC↑	WRAP_E↓	MD↓	FID↓	FID_{dino}↓	KD↓
Bilinear (released)	0.9661	0.0181	3.889	200.33	2449.63	0.0231
Exact integer shift	0.9661	0.0181	3.889	200.33	2449.63	0.0231
Anti-aliased	0.9659	0.0211	3.805	200.51	2449.70	0.0235

Kernel distance is seeded.

11.5 Experiment 16: What makes interior restoration work

11.5.1 Change

Two controls isolate the two properties that could explain the gains.

- **Scope.** The same operation was applied either to translations only or to every edit: latent preservation at $w = 0.3$ (Experiment 12), and interior restoration with $w = 8$. The global version of latent preservation reproduces the numbers of Experiment 12 exactly on every deterministic metric, which confirms the mechanism across two separate studies.

- **Warp quality.** The composite of every translation was rebuilt in two further ways (by an exact integer shift, and with the anti-aliased warp of Experiment 7), the translations were regenerated from it, and interior restoration with $w = 8$ was applied.

11.5.2 Results

Table 11.5 shows that the versions restricted to translations leave rotations and resizings exactly as they were, while the global versions damage them. Latent preservation on all edits worsens mean distance on resizings by 1.55 pixels and every realism metric of the whole subset, FID by 0.86 and the kernel distance by 0.011. Interior restoration on all edits raises subject consistency on the whole subset the most, by 0.0053, but worsens mean distance on rotations by 1.67 pixels and their FID by 0.59, while it slightly improves resizings. In Table 11.6 the exact integer shift reproduces the released warp to every digit, and the anti-aliased composite is marginally worse on subject consistency, warping

Table 11.7: Interior restoration applied to each edit type: baseline against restored.

			SUBC↑		WRAP_E↓		MD↓	
Group	n	w	baseline	restored	baseline	restored	baseline	restored
Translations	67	4	0.9590	0.9670	0.0495	0.0115	3.41	3.96
Resizings, easy and medium	44	8	0.9176	0.9237	0.0523	0.0224	7.58	7.32
Resizings, hard	23	8	0.8391	0.8386	0.0585	0.0259	19.47	18.97
Rotations	66	8	0.8962	0.9012	0.0423	0.0176	9.63	11.30
Rotations, hard	22	8	0.8523	0.8574	0.0416	0.0112	14.01	17.92

FID within each group, baseline against restored: 200.04 against 199.99 (translations), 221.85 against 221.95 (easy and medium resizings), 274.34 against 273.55 (hard resizings), 205.96 against 206.55 (rotations). Kernel distance is seeded.

error and FID and marginally better on mean distance.

11.5.3 Outcome

Scope decides. The same operation helps the edit types it suits and harms the others, so it has to be applied selectively. Latent preservation belongs, if anywhere, to translations; interior restoration harms rotations, as Experiment 14 found, and does not harm resizings.

Restoration works because the interior is exact. Every translation of GeoBench is a whole number of pixels, so the bilinear warp already copies the object exactly, and an exact shift changes nothing. The anti-aliased warp filters an already exact copy, which makes the restored interior slightly less faithful. Better resampling cannot improve a composite that involves no resampling: restoration gains because it keeps source pixels that are already correct.

Next. Interior restoration is useful where the composite's interior is reliable, and that depends on the edit type. The last experiment applies it to each type separately.

11.6 Experiment 17: Interior restoration for each edit type

11.6.1 Change

Interior restoration was applied to each edit type separately: to translations with the width of Experiment 15, $w = 4$, and to rotations and resizings with $w = 8$, the width tested on every edit type in Experiment 16. Resizings are divided by severity into the hard tertile and the easy and medium tertiles together (§7.2.1), because a resized object is resampled more strongly the further its scale is from one.

11.6.2 Results

Table 11.7 reports interior restoration for each type. On translations and on easy and medium resizings it raises subject consistency, by 0.0080 and 0.0061, and lowers the warping error by 77% and 57%; mean distance worsens by 0.55 pixels on translations and improves by 0.26 on the resizings. On hard resizings subject consistency does not move (-0.0005) and mean distance falls by 0.50 pixels. On rotations subject consistency rises by 0.0050 and 0.0051, and mean distance worsens by 1.67 and 3.91 pixels.

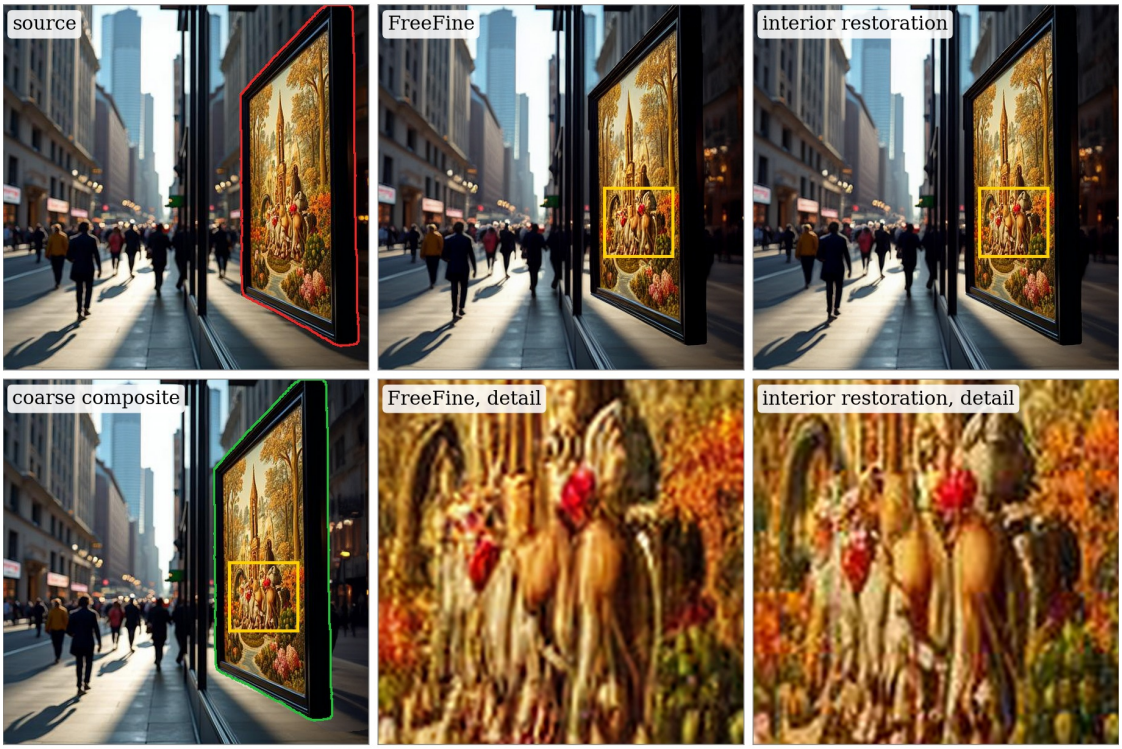
Figure 11.4 shows what restoration changes in the image. On a translated painting and on a resized jar of cream, FreeFine redraws the fine detail of the object (the figures of the painting and the lettering on the jar, which becomes illegible), while interior restoration keeps the composite's copy of the source and refines only the band along the outline.

11.6.3 Outcome

Three regimes. Interior restoration is worth applying where the composite's interior is reliable: translations, where it is an exact copy, and moderate resizings, where the resampled interior is still close to the source and a wider band gives the boundary room to blend. It is harmful on rotations: subject consistency rises only by what the copied pixels contribute, while mean distance worsens by several times what restoration costs on translations, the same pattern as in Experiment 14. And it gains nothing on hard resizings, whose interior is resampled so strongly that it offers little worth keeping; there the edit has to be reconstructed rather than preserved.

What the chapter establishes. Hypothesis 4 holds selectively. Refinement does lose information that the composite contains, but little of it on subject consistency (the refined output is within one per cent of the composite), and keeping the composite pays only where its interior is reliable. There, a thin refined band around a restored interior improves consistency on unseen edits without costing realism; elsewhere it trades one metric for another or harms the edit. The choice must therefore depend on the edit: on its type, and for resizings on how strongly the object is rescaled. Severe resizings remain the one case that preservation cannot help, and the one case where guidance with the object label already helps (Experiment 5). Chapter 12 develops that guidance into a reconstruction method for severe resizings.

(a) Translation: 37 pixels to the left, $w = 4$



(b) Moderate resizing: to 0.85 of the original size, $w = 8$



Figure 11.4: FreeFine against interior restoration on (a) a translation ($w = 4$) and (b) a moderate resizing ($w = 8$). In each part, the top row shows the source image (red: the object's mask) and the whole outputs; the bottom row shows the coarse composite (green: the target mask) and the region in the yellow box, enlarged.

Chapter 12

Hypothesis 5: Guiding the Reconstruction of Severe Resizings

Chapter 11 ended with one case that anchoring cannot help. On hard resizings the composite's interior is resampled so strongly that restoring it gains nothing (Experiment 17), and yet the composite places the object's keypoints better than the refined output, 16.44 against 19.47 pixels (Experiment 13): refinement moves the resized content away from where the transformation put it. A severe resizing asks refinement to reconstruct the object at a scale at which the source never showed it, and the only change so far that has reduced the loss is guidance with the object label (Experiment 5), which lowers mean distance on hard resizings by 1.73 pixels while acting only on the band around the object. This chapter asks whether that reconstruction can be steered better.

> **Hypothesis 5.** On severe resizings refinement must reconstruct the object's structure at a new scale, and it does so more accurately when the guidance that steers this reconstruction is given the right form and applied where it is needed.

Target and criterion. The hypothesis speaks of severe resizings, those whose scale is far from one. The term receives its exact definition only with the routing rule of Experiment 24 (Eq. (14.1)); until then the group on which it is measured is the hard tertile of the resizings (§8.2), the hard resizings. Subject consistency on hard resizings is within 0.2% of the composite's (0.8391 against 0.8408, Experiment 13), so there is almost nothing to gain on it. The chapter is judged by mean distance on the 23 hard resizings of the development subset, against label guidance at $w = 10$ (17.75 pixels). A change counts if it lowers mean distance clearly more than the one to two pixels that separate near-identical variants (Experiment 3), or if a sequence of related changes lowers it consistently, without lowering subject consistency, background consistency or realism.

Where the change acts. One refined step of the edit branch offers two places to steer the reconstruction (Figure 12.1). This chapter uses the first, the *guidance combination* after the U-Net, where the unguided and the label-conditioned noise predictions are combined inside the band m_{cfg} around the object (Eq. (5.9)). Experiments 18 to 21 change the form of this combination: projected guidance on the noise predictions, in ε-space (Experiment 18), guidance split by spatial frequency (Experiment 19), projected guidance on the denoised predictions, in D-space, which yields the EP-SREC formulation (Experiment 20), and the scope over which guidance is applied (Experiment 21). The second place, the *decoder features* inside the object, which no guidance reaches, is the subject of Chapter 13.

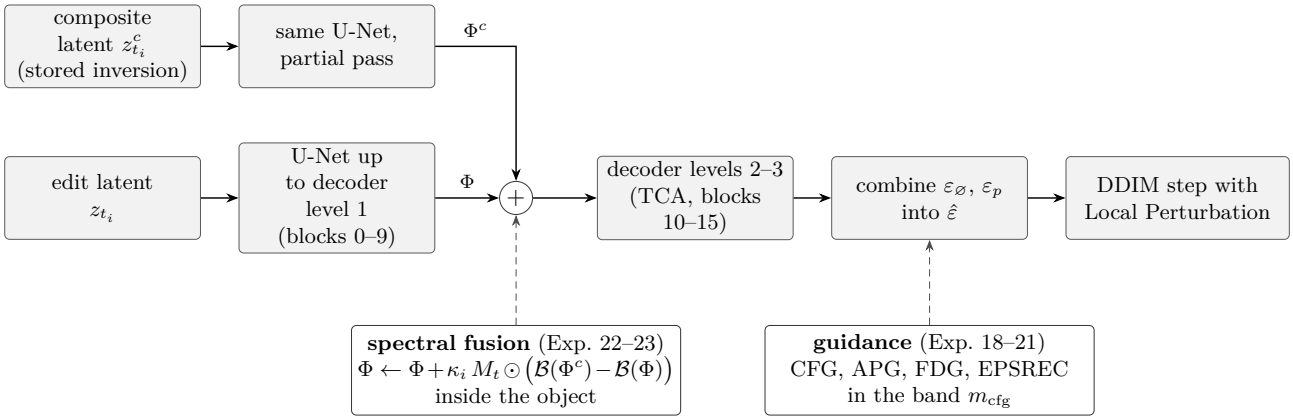


Figure 12.1: One refined step of the edit branch and the two points at which the reconstruction is steered. Guidance, the subject of this chapter, changes how the unguided and the label-conditioned noise predictions are combined, in the band m_{cfg} around the object. Spectral fusion, the subject of Chapter 13, changes the decoder features Φ inside the object, using the features Φ^c that the same U-Net computes from the composite's stored inversion latent. The reference branches are not changed.

Subsets. Experiments 18 to 20 regenerate only the 67 resizings of the development subset, because the mechanisms are switched on only for them; every other edit would be identical to the model it is compared with. They report subject consistency, background consistency, the warping error and mean distance on all resizings and on the hard resizings. Experiment 21 regenerates all 200 edits and reports every metric, with the kernel distance seeded. The reference model is regenerated in every study and reproduces the same values to all four decimals each time.

12.1 Experiment 18: Projected guidance in ε-space

12.1.1 Change

The guidance correction. Write $\Delta_i = \varepsilon_p - \varepsilon_\varnothing$ for the correction that the object label adds to the noise prediction at refined step i. Inside m_{cfg}, Eq. (5.8) replaces the prediction by

$$\hat{\varepsilon} = \varepsilon_\varnothing + w\,\Delta_i = \varepsilon_p + (w-1)\,\Delta_i, \tag{12.1}$$

so at $w = 10$ the prediction is extrapolated nine corrections beyond the conditional one. Large scales are known to over-saturate colours and exaggerate contrast (§3.7), and Sadat et al. [42] traced these artefacts to one component of the correction. Relative to a vector a, a correction d splits into a part along a and a part orthogonal to it,

$$d = \mathrm{P}_a d + (d - \mathrm{P}_a d), \qquad \mathrm{P}_a d = \frac{\langle d, a\rangle}{\langle a, a\rangle}\, a, \tag{12.2}$$

with inner products and norms taken over the whole latent of a branch. Taken relative to the conditional prediction, the parallel part only rescales what the model already predicts; the orthogonal part changes what it predicts. Adaptive Projected Guidance (APG) keeps the orthogonal part at full weight, weights the parallel part by $\nu \le 1$, limits the norm of the correction to a threshold r, and adds a momentum term with a negative coefficient μ that subtracts part of the previous correction, so that the same push is not applied again and again.

Table 12.1: Projected guidance in ε-space, Eq. (12.3), against label guidance, with fifteen ($s = 35$) and five ($s = 45$) refined steps.

	All resizings ($n = 67$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Released (no guidance)	0.8906	0.0544	10.93	0.8391	0.0585	19.47
Label, CFG $w = 10$	0.8908	0.0547	11.28	0.8391	0.0587	17.75
Label, CFG, $s = 45$	0.8929	0.0502	11.22	0.8416	0.0536	18.08
Label, APG	0.8908	0.0545	11.11	0.8391	0.0585	17.30
Label, APG, $s = 45$	0.8929	0.0501	11.27	0.8416	0.0536	18.41

Background consistency on hard resizings: 0.9623–0.9631.

The rule in ε-space. FreeFine combines noise predictions, and the first question is whether APG's operations help when applied to them directly, in the space of the noise predictions, called ε-space in this chapter:

$$\begin{aligned} d_i &= (\Delta_i + \mu\,\Delta_{i-1})\,\min\Big(1, \frac{r}{\|\Delta_i + \mu\,\Delta_{i-1}\|}\Big), \qquad \Delta_{s-1} = 0, \\ \varepsilon^g &= \varepsilon_p + (w-1)\big[(d_i - \mathrm{P}_{\varepsilon_p} d_i) + \nu\,\mathrm{P}_{\varepsilon_p} d_i\big], \qquad \hat{\varepsilon} = \varepsilon_\varnothing + (\varepsilon^g - \varepsilon_\varnothing) \odot m_{\mathrm{cfg}}. \end{aligned} \tag{12.3}$$

With $\mu = 0$, no threshold and $\nu = 1$ the rule reduces to Eq. (5.8). It replaces the single line of the sampling loop that forms the guided prediction; the four-branch batch, the regions and the rest of the step are unchanged, and the momentum is reset for every edit. The settings are $w = 10$, $r = 5$, $\mu = -0.5$ and $\nu = 0$ (η in [42]), which removes the parallel part entirely, as the APG paper recommends.

Refinement strength. Experiment 3 found that starting refinement later, at $s = 45$ instead of 35, lowers mean distance on hard resizings to 18.09 by keeping more of the composite. Label guidance and APG were therefore also run at $s = 45$, with five refined steps instead of fifteen, to test whether weaker refinement and guidance add up.

12.1.2 Results

Table 12.1 reports the four variants. APG lowers mean distance on hard resizings to 17.30, 0.45 pixels below label guidance, and on all resizings to 11.11, while subject consistency and the warping error stay where label guidance leaves them. At $s = 45$ subject consistency on hard resizings rises by 0.0025 and the warping error falls by 0.005 for both kinds of guidance, but mean distance is worse than with fifteen steps: 18.08 with label guidance, practically the 18.09 of $s = 45$ without any guidance, and 18.41 with APG.

12.1.3 Outcome

The right direction, below resolution. Removing the parallel part and damping repeated pushes keeps the geometric effect of guidance and adds a little to it. The 0.45 pixels are below what mean distance resolves on 23 edits, so this experiment alone is not evidence that APG helps; it shows that the form of the correction, not only its scale, can be changed without losing what the correction does.

Guidance needs the refinement window. With five refined steps label guidance changes nothing (18.08 against 18.09), and APG does not help either. The gains in subject consistency and warping error at $s = 45$ are those of copying more of the composite, as in Experiment 3. Steering a reconstruction requires steps in which to reconstruct, and weakening refinement cannot be combined with steering it; the start step stays at 35.

Next. Before APG is examined in the space for which it was designed, the next experiment tests a different way of splitting the correction: by spatial frequency instead of by direction.

12.2 Experiment 19: Frequency-decoupled guidance

12.2.1 Change

From direction to frequency. Experiment 18 split the guidance correction by direction, into a part along the conditional prediction and a part orthogonal to it. This experiment splits the same correction by spatial frequency, into its broad and its fine variations, and asks which of the two carries the geometric effect that label guidance has on hard resizings.

The spatial frequencies of the correction. The correction Δ_i is a map: four channels over the 64×64 latent grid, each latent pixel covering 8×8 image pixels. Like any image, it can be written as a sum of periodic patterns. A pattern that repeats every λ latent pixels has the frequency $f = 1/\lambda$ cycles per latent pixel. The finest pattern the grid can hold alternates between neighbouring latent pixels, $f = 0.5$, a wavelength of 16 image pixels; the broadest vary only once across the whole map. Low frequencies therefore describe how the correction changes over large distances (which parts of the band around the object are pushed towards the label, where regions and outlines lie), and high frequencies how it changes between neighbouring positions: edges, texture and fine detail. Classifier-free guidance multiplies every frequency of the correction by the same scale w.

What the two bands do. Sadat et al. [43] analysed classifier-free guidance in the frequency domain and found that the two bands affect generation differently: low-frequency guidance governs the global structure and the agreement with the condition, high-frequency guidance mainly the fidelity of detail, and one scale for both leads to over-saturation and reduced diversity when the scale is high. Their frequency-decoupled guidance (FDG) splits the correction into a low and a high band and gives each band its own scale.

The rule. The split uses a 5×5 binomial low-pass filter, applied to each channel with reflected borders:

$$g = \tfrac{1}{256}[1\,4\,6\,4\,1]^\top [1\,4\,6\,4\,1], \qquad \Delta_i^{\mathrm{lo}} = g * \Delta_i, \qquad \Delta_i^{\mathrm{hi}} = \Delta_i - \Delta_i^{\mathrm{lo}},$$
$$\varepsilon^g = \varepsilon_p + (w_{\mathrm{lo}} - 1)\,\Delta_i^{\mathrm{lo}} + (w_{\mathrm{hi}} - 1)\,\Delta_i^{\mathrm{hi}}, \tag{12.4}$$

followed by the regional combination of Experiment 18, $\hat{\varepsilon} = \varepsilon_\varnothing + (\varepsilon^g - \varepsilon_\varnothing) \odot m_{\mathrm{cfg}}$ (Figure 12.2a). The two bands add up to the correction, so the guided correction inside the region is $\varepsilon^g - \varepsilon_\varnothing = w_{\mathrm{lo}}\Delta_i^{\mathrm{lo}} + w_{\mathrm{hi}}\Delta_i^{\mathrm{hi}}$. Along either axis of the map the filter passes a pattern of frequency f with the

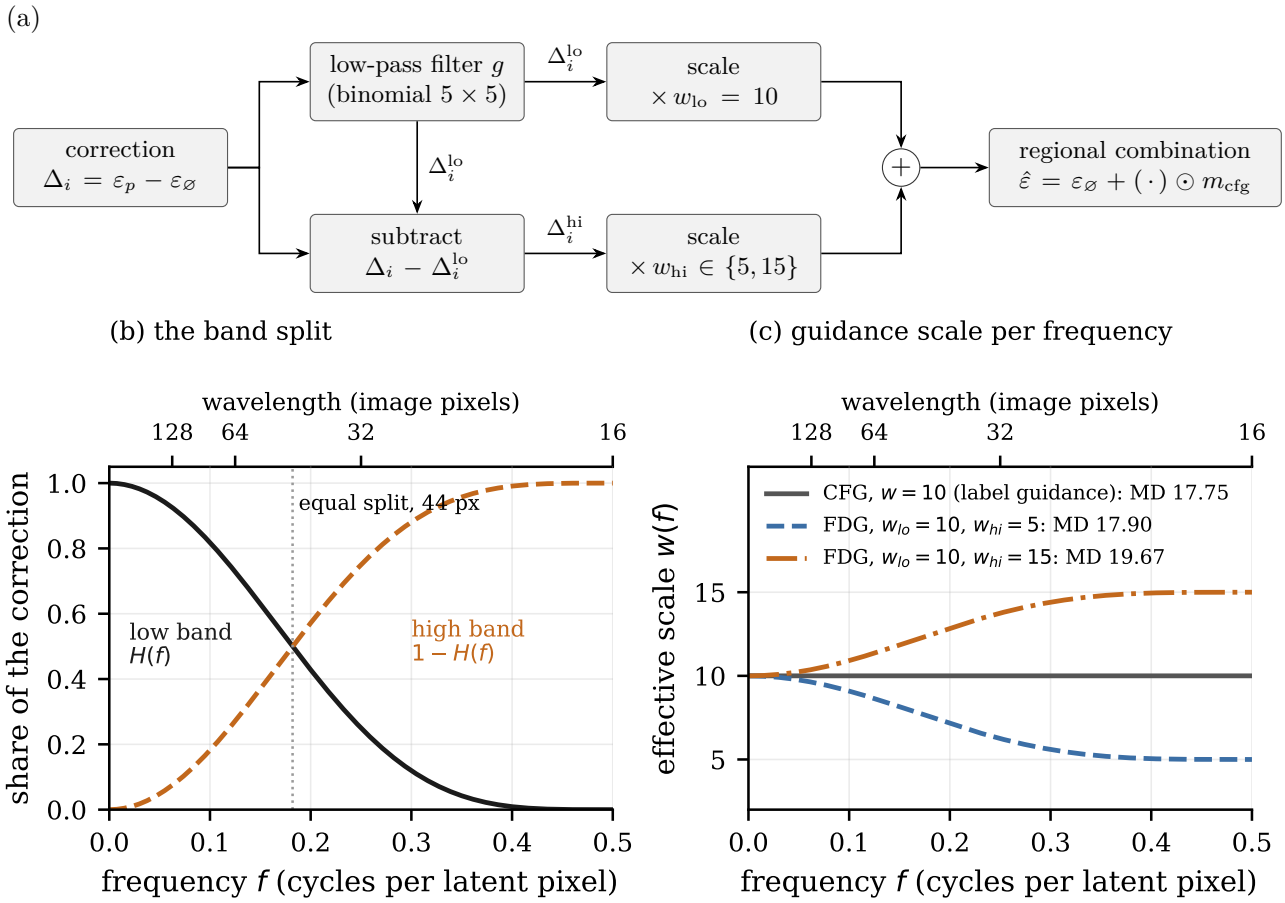


Figure 12.2: Frequency-decoupled guidance. (a) The label's correction is split by a low-pass filter into a broad and a fine band, each band is scaled separately, and their sum replaces the correction of label guidance inside m_{cfg}. (b) The share of each band in a pattern of spatial frequency f, along either axis of the latent map; the top axis gives the corresponding wavelength in image pixels. (c) The resulting guidance scale per frequency, Eq. (12.5), for label guidance and the two settings tested, with the mean distance each obtains on hard resizings.

factor $H(f) = \cos^4(\pi f)$, and FDG is therefore classifier-free guidance whose scale depends on the frequency:

$$w(f) = w_{\text{lo}}\, H(f) + w_{\text{hi}}\, [1 - H(f)]. \tag{12.5}$$

The share $H(f)$ of the low band falls from 1 for the broadest patterns to 0 for the finest, and the two bands share a pattern equally at $f = 0.18$, a wavelength of about 5.5 latent or 44 image pixels (Figure 12.2b). Broader patterns are guided mostly with w_{lo}, finer ones mostly with w_{hi} (Figure 12.2c). With $w_{\text{lo}} = w_{\text{hi}} = w$ the scale is w at every frequency and the rule is Eq. (5.8).

Two settings and what they can show. The low band keeps the scale at which label guidance helps, $w_{\text{lo}} = 10$ (Experiment 5), and only the high band is changed: halved to $w_{\text{hi}} = 5$, or raised by half to $w_{\text{hi}} = 15$. The two settings separate three possible roles of the high band.

- If the geometric effect of guidance lies in the low band, neither setting moves mean distance on hard resizings away from the 17.75 of label guidance.

- If high-frequency detail helps the keypoints find the object, $w_{\text{hi}} = 15$ lowers mean distance and

Table 12.2: Frequency-decoupled guidance, Eq. (12.4), with the low band at the scale of label guidance.

	All resizings ($n = 67$)			Hard resizings ($n = 23$)		
	SUBC↑	WRAP_E↓	MD↓	SUBC↑	WRAP_E↓	MD↓
Released (no guidance)	0.8906	0.0544	10.93	0.8391	0.0585	19.47
Label guidance, $w_{\text{lo}} = w_{\text{hi}} = 10$	0.8908	0.0547	11.28	0.8391	0.0587	17.75
FDG, $w_{\text{lo}} = 10$, $w_{\text{hi}} = 5$	0.8908	0.0545	11.35	0.8392	0.0585	17.90
FDG, $w_{\text{lo}} = 10$, $w_{\text{hi}} = 15$	0.8909	0.0549	11.96	0.8393	0.0589	19.67

Background consistency on hard resizings: 0.9625–0.9631.

$w_{\text{hi}} = 5$ raises it.

- If high-frequency guidance adds artefacts that mislead the keypoints, $w_{\text{hi}} = 5$ lowers mean distance.

The rule replaces the same line of the sampling loop as Experiment 18 (Figure 12.1) and is applied to the 67 resizings; everything else is unchanged, and $w_{\text{lo}} = w_{\text{hi}} = 10$ reproduces label guidance.

12.2.2 Results

Table 12.2 and Figure 12.2c report the two settings, and the result matches the first of the three possibilities. Halving the high band leaves mean distance on hard resizings at 17.90, within resolution of the 17.75 of label guidance, and on all resizings at 11.35 against 11.28. Raising it by half removes the entire gain of guidance: mean distance on hard resizings rises to 19.67, above the 19.47 of the released pipeline, and on all resizings to 11.96, 0.69 pixels worse than with label guidance. Subject consistency, background consistency and the warping error do not move in either setting.

12.2.3 Outcome

The effect of guidance lives in the low band. Guidance at the full scale on the patterns broader than about 44 pixels, and at half the scale on the finer ones, does what label guidance does. The part of the correction that moves resized content towards its place is its low-frequency part: the broad push that rebuilds the band around the object as the surroundings of the labelled object. This agrees with the observation of Sadat et al. that low-frequency guidance governs structure and the agreement with the condition.

Fine detail in the wrong place hurts. Raising the high band does not merely fail to help; it cancels the benefit of guidance completely. At $w_{\text{hi}} = 15$ the finest patterns of the correction are applied at one and a half times the scale, which adds edges and texture to the band around the object, along its new outline and over the area it vacated. Residual structure of this kind is what a correspondence-based metric can match to (Experiment 5): the keypoint matcher finds features of the object in its surroundings instead of on the object. The object itself is left as it was (subject consistency and the

warping error, both measured inside it, do not change), so what worsens mean distance lies in the surroundings, where the guidance acts.

Why frequency decoupling is not pursued. Beyond classifier-free guidance the split adds one free choice, the scale of the high band, and neither direction helps: lowering it changes nothing and raising it undoes the gain. The low band carries the effect and is already at the scale that works; lowering it would weaken that effect, as the dose response of label guidance indicates (18.38 at $w = 7.5$ against 17.75 at $w = 10$, Experiment 5). Which frequencies of the correction are applied is therefore not where guidance can be improved. What remains open is how the correction is applied across the refined steps, that is, the momentum that Experiment 18 introduced, and the next experiment examines it in the space for which APG was designed. The frequency view returns in Chapter 13, applied to the object's features instead of the guidance.

12.3 Experiment 20: APG in D-space and the EPSREC formulation

12.3.1 Change

Experiment 18 applied the three operations of APG (projection, norm threshold and momentum) to the noise predictions. APG was designed, however, for a different quantity: the network's prediction of the clean image. This experiment moves APG to that quantity and answers two questions. Does it matter which of the two quantities APG operates on? And which of its three operations produces the gain? The answer is a formulation called EPSREC, which is the guidance used for severe resizings in the rest of the thesis. The change is explained in six steps: the two spaces (1), why plain guidance does not depend on the space (2), the three operations of APG one at a time (3–5), and the variants that were run together with EPSREC itself (6).

1. Two spaces for the same prediction

At refined step i, with timestep t_i, the U-Net returns two noise predictions for the edit branch: the unguided $\varepsilon_\varnothing$ and the label-conditioned ε_p. The space of these noise predictions is called *ε-space*; it is the noise space of §3.7.

Every noise prediction can be turned into a prediction of the clean latent: the latent the sampler would reach if it removed all the predicted noise at once (Eq. (3.7)). Following Sadat et al. [42], this *denoised prediction* is written D:

$$D = \frac{z_{t_i} - \sqrt{1-\bar{\alpha}_{t_i}}\,\varepsilon}{\sqrt{\bar{\alpha}_{t_i}}}, \qquad \text{giving } D_\varnothing \text{ from } \varepsilon_\varnothing \text{ and } D_p \text{ from } \varepsilon_p. \tag{12.6}$$

The space of the denoised predictions is called *D-space*; it is the clean-sample space of §3.7. Unlike a noise prediction, D is itself an image, in latent form, and statements such as "raise the contrast" have a meaning for it.

The label's correction exists in both spaces:

$$\Delta\varepsilon_i = \varepsilon_p - \varepsilon_\varnothing, \qquad \Delta D_i = D_p - D_\varnothing. \tag{12.7}$$

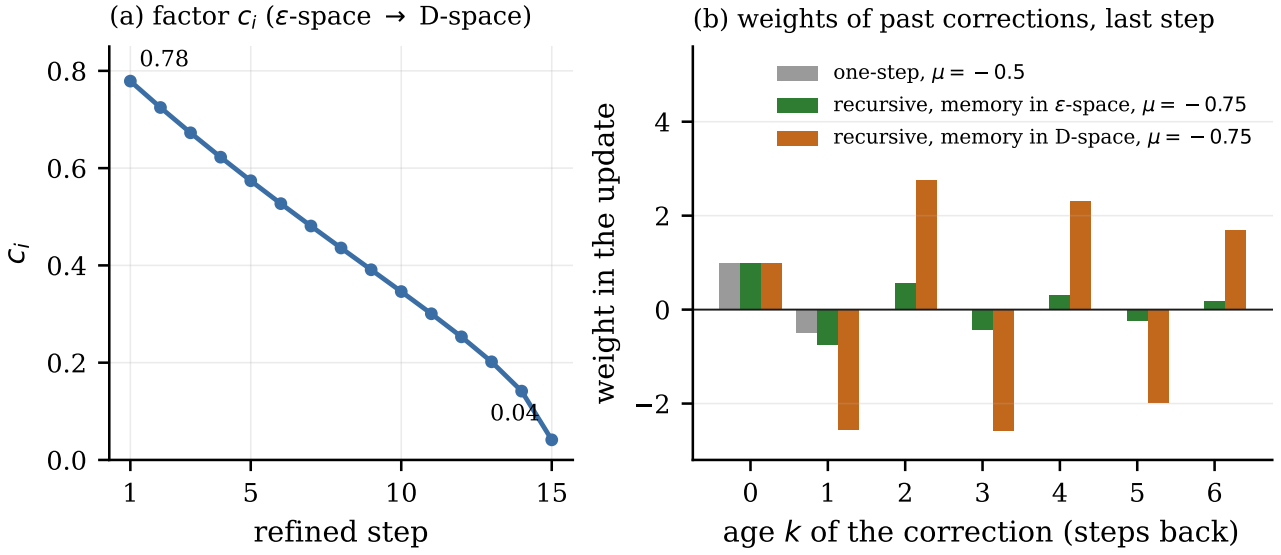


Figure 12.3: (a) The factor c_i that converts an ε-space correction into a D-space correction, Eq. (12.8), at the fifteen refined steps ($t = 281$ to $t = 1$). (b) The weight with which the correction made k steps earlier enters the momentum at the last refined step, for one-step momentum ($\mu = -0.5$) and for recursive momentum ($\mu = -0.75$) with the memory stored in ε-space or in D-space.

$\Delta\varepsilon_i$ is the correction Δ_i of Experiment 18. Writing Eq. (12.6) for D_p and for $D_\varnothing$ and subtracting, the latent z_{t_i} cancels and

$$\Delta D_i = -\frac{\sqrt{1-\bar\alpha_{t_i}}}{\sqrt{\bar\alpha_{t_i}}}\,\Delta\varepsilon_i = -c_i\,\Delta\varepsilon_i, \qquad c_i = \sqrt{\frac{1-\bar\alpha_{t_i}}{\bar\alpha_{t_i}}}, \tag{12.8}$$

where c_i is the factor c_t of Eq. (3.11) at the timestep of step i. The two corrections point in opposite directions of the same line and differ in size by the factor c_i. That factor depends strongly on the step (Figure 12.3a): $c_i = 0.78$ at the first refined step ($t = 281$) and $c_i = 0.04$ at the last ($t = 1$), about nineteen times smaller. The same correction of the noise prediction therefore changes the predicted image nineteen times more at the beginning of refinement than at the end.

2. Plain guidance is the same in both spaces

Classifier-free guidance in ε-space extrapolates the conditional prediction by the correction, Eq. (12.1):

$$\varepsilon^g = \varepsilon_p + (w-1)\,\Delta\varepsilon_i.$$

The same rule in D-space is $D^g = D_p + (w-1)\,\Delta D_i$. To be used by the sampler, D^g has to be converted back into a noise prediction by inverting Eq. (12.6):

$$\varepsilon^g = \frac{z_{t_i} - \sqrt{\bar\alpha_{t_i}}\,D^g}{\sqrt{1-\bar\alpha_{t_i}}}. \tag{12.9}$$

Substituting $D^g = D_p + (w-1)\Delta D_i$ and $\Delta D_i = -c_i\Delta\varepsilon_i$, the first part returns ε_p and the second part returns $(w-1)\Delta\varepsilon_i$:

$$\varepsilon^g = \underbrace{\frac{z_{t_i} - \sqrt{\bar\alpha_{t_i}}\,D_p}{\sqrt{1-\bar\alpha_{t_i}}}}_{=\varepsilon_p} + (w-1)\,\underbrace{\frac{\sqrt{\bar\alpha_{t_i}}}{\sqrt{1-\bar\alpha_{t_i}}}\,c_i}_{=1}\,\Delta\varepsilon_i = \varepsilon_p + (w-1)\,\Delta\varepsilon_i. \tag{12.10}$$

Both routes give the same prediction. Plain guidance only scales the correction, and scaling commutes with the factor c_i. APG does more than scale: it projects, caps and accumulates the correction, and each of these operations is affected by c_i differently. The next three steps take them one at a time.

3. Operation one: the projection

APG splits the correction into a part along the conditional prediction and a part orthogonal to it, Eq. (12.2), and removes the parallel part ($\nu = 0$). The conditional prediction is different in the two spaces:

- in ε-space (Experiment 18) the parallel part is taken along ε_p;
- in D-space it is taken along D_p.

By Eq. (12.6), D_p is a mixture of the current latent and the noise prediction, not a multiple of ε_p: at the first refined step $D_p = 1.27\, z_{t_i} - 0.78\, \varepsilon_p$, and at the last $D_p \approx z_{t_i}$. The two projections therefore remove different parts of the correction.

Only in D-space does the parallel part have a clear meaning. A correction along D_p changes the predicted image by a multiple of itself: it makes the image stronger or weaker, in contrast and saturation, without changing what the image shows. Sadat et al. identified this part as the cause of over-saturation at high guidance scales, and APG removes it.

4. Operation two: the norm threshold

APG caps the size of the correction at a threshold r, $d \leftarrow d \cdot \min(1,\, r/\|d\|)$, where the norm is taken over the whole latent of the branch. By Eq. (12.8), $\|\Delta D_i\| = c_i \, \|\Delta\varepsilon_i\|$, so a threshold means different things in the two spaces.

- A *fixed* D-space threshold r_D is the ε-space threshold r_D/c_i, which changes from step to step. For $r_D = 7.5$ it is 9.6 at the first refined step and 182 at the last.
- To keep the ε-space threshold of Experiment 18, $r = 5$, at every step, the D-space threshold has to follow c_i: $r_D(i) = 5\, c_i$, from 3.89 at the first step to 0.21 at the last. This is called the *ε-equivalent threshold*.

5. Operation three: the momentum

The momentum has two separate aspects: the rule by which past corrections are combined, and the space in which they are stored.

The rule: one-step or recursive. The *one-step* rule of Experiment 18 adds a fraction of the previous correction only:

$$d_i = \Delta\varepsilon_i + \mu\,\Delta\varepsilon_{i-1}, \qquad \mu = -0.5.$$

The *recursive* rule keeps a running memory u_i of all past corrections, starting from zero at the first refined step s:

$$u_i = \Delta\varepsilon_i + \mu\,u_{i-1}, \quad u_{s-1} = 0 \qquad \Longrightarrow \qquad u_i = \Delta\varepsilon_i + \mu\,\Delta\varepsilon_{i-1} + \mu^2\,\Delta\varepsilon_{i-2} + \dots \tag{12.11}$$

The correction made k steps earlier enters with the weight μ^k. With $\mu = -0.75$ the weights are 1, -0.75, 0.56, -0.42, 0.32, and so on: each step subtracts part of the previous correction and adds back part of the one before.

What a negative μ does is clearest in two extreme cases.

- If the correction is the same at every step, $\Delta\varepsilon_i = \Delta$, the memory settles at $\Delta/(1-\mu) = 0.57\,\Delta$. A push that guidance keeps repeating is damped.

- If the correction flips sign at every step, the memory settles at $\Delta/(1+\mu) = 4\,\Delta$ in size. A correction that changes is passed on, even amplified.

Recursive momentum with $\mu < 0$ therefore weakens the part of the correction that persists from step to step and keeps the part that changes.

The space: where the memory is stored. The recursion can be run on either correction of Eq. (12.7).

- **Memory in D-space:** $u_i^D = \Delta D_i + \mu\,u_{i-1}^D$. Substituting $\Delta D = -c\,\Delta\varepsilon$ at every step and unrolling,

$$u_i^D = -(c_i\,\Delta\varepsilon_i + \mu\,c_{i-1}\,\Delta\varepsilon_{i-1} + \mu^2\,c_{i-2}\,\Delta\varepsilon_{i-2} + \dots) = -c_i \sum_{k\geq 0} \mu^k\,\frac{c_{i-k}}{c_i}\,\Delta\varepsilon_{i-k}. \tag{12.12}$$

  Relative to the current step, the correction made k steps earlier has the weight $\mu^k\,c_{i-k}/c_i$. Since c decreases along the window, $c_{i-k}/c_i > 1$: older corrections are inflated, and the more so the later the step.

- **Memory in ε-space:** $u_i = \Delta\varepsilon_i + \mu\,u_{i-1}$, Eq. (12.11), converted to D-space only once, with the current factor: $d_i = -c_i\,u_i$. The weights are μ^k at every step.

Table 12.3 gives the weights at three points of the window, and Figure 12.3b shows them at the last step. Storing the memory in D-space mixes corrections measured on different scales; storing it in ε-space and converting once keeps a single scale.

6. The variants and EPSREC

The variants. Five variants were run, all with $w = 10$ and $\nu = 0$. Table 12.4 lists their settings next to those of Experiment 18.

Each comparison isolates one question.

Table 12.3: Weight of the correction made k steps earlier in the recursive memory, $\mu = -0.75$.

	$k=0$	$k=1$	$k=2$	$k=3$
Memory in ε-space, any step	1	−0.75	0.56	−0.42
Memory in D-space, 3rd refined step ($t=241$)	1	−0.81	0.65	n/a
Memory in D-space, 8th refined step ($t=141$)	1	−0.83	0.68	−0.56
Memory in D-space, last refined step ($t=1$)	1	−2.57	2.75	−2.59

D-space weights are relative to the current step's scale, $\mu^k\, c_{i-k}/c_i$, Eq. (12.12). At the 3rd refined step only two earlier corrections exist.

Table 12.4: Settings of the APG variants. All use $w = 10$ and $\nu = 0$.

Variant	Projection along	Threshold	Momentum rule	μ	Memory stored in
Experiment 18	ε_p	5 in ε-space	one-step	−0.5	none
(a)	D_p	$5c_i$ (ε-equivalent)	one-step	−0.5	none
(b)	D_p, inside m_{cfg}	$5c_i$ (ε-equivalent)	one-step	−0.5	none
(c)	D_p	$5c_i$ (ε-equivalent)	recursive	−0.75	D-space
(d)	D_p	7.5 (fixed)	recursive	−0.75	D-space
(e) EPSREC	D_p	$5c_i$ (ε-equivalent)	recursive	−0.75	ε-space

- **(a) against Experiment 18: the projection.** With one-step momentum, the D-space correction of (a) is the correction of Experiment 18 multiplied by $-c_i$, and the ε-equivalent threshold caps it exactly as before. Only the vector along which the parallel part is removed differs: D_p instead of ε_p.

- **(b) against (a): where the projection statistics are computed.** The correction is applied only inside the band m_{cfg}, so in (b) the inner products of the projection and the norm of the threshold are computed inside the band only. The norm is divided by the square root of the band's area fraction, so that the threshold keeps its meaning.

- **(c) against (a): the momentum rule.** Recursive instead of one-step, with the coefficient $\mu = -0.75$ used with the recursive rule instead of -0.5. These two settings change together, and their joint effect is measured.

- **(d) against (c): the threshold.** A fixed D-space threshold of 7.5 instead of the ε-equivalent $5c_i$.

- **(e) against (c): where the memory is stored.** In ε-space instead of D-space; nothing else differs.

EPSREC. Variant (e) is called EPSREC: APG with a recursive memory stored in ε-space. At every refined step i it computes the guided noise prediction in eight steps:

1. the label's correction in ε-space, $\Delta\varepsilon_i = \varepsilon_p - \varepsilon_\varnothing$;

2. the memory in ε-space, $u_i = \Delta\varepsilon_i + \mu\, u_{i-1}$, with $u_{s-1} = 0$ and $\mu = -0.75$;

3. one conversion to D-space with the current factor, $d_i = -c_i\, u_i$;

Table 12.5: Mean distance on the resizings for the APG variants of Table 12.4.

	Mean distance↓		
Variant	All ($n = 67$)	Easy, medium ($n = 44$)	Hard ($n = 23$)
Released (no guidance)	10.93	7.58	19.47
Label guidance, CFG $w = 10$	11.28	7.80	17.75
Experiment 18: APG in ε-space	11.11	n/a	17.30
(a) projection along D_p	11.10	7.81	17.39
(b) statistics inside m_{cfg}	11.15	7.88	17.31
(c) recursive, memory in D-space	11.00	7.87	16.77
(d) as (c), fixed threshold 7.5	11.00	7.87	16.77
(e) EPSREC: memory in ε-space	10.92	7.78	16.61
EPSREC on hard resizings only	10.81	7.58	16.61

On hard resizings all variants (a)–(e) give subject consistency 0.8390–0.8391, background consistency 0.9625–0.9627 and warping error 0.0585. The study of Experiment 18 did not score the easy and medium resizings separately.

4. the ε-equivalent threshold, $d_i \leftarrow d_i \cdot \min(1,\, 5c_i/\|d_i\|)$, which is the same as capping $\|u_i\|$ at 5;

5. the projection along D_p, $d_{\|} = \mathrm{P}_{D_p} d_i$ and $d_{\perp} = d_i - d_{\|}$, keeping $d_{\text{mod}} = d_{\perp} + \nu\, d_{\|} = d_{\perp}$ since $\nu = 0$;

6. the guided denoised prediction, $D^g = D_p + (w - 1)\, d_{\text{mod}}$, with $w = 10$;

7. the conversion back to ε-space by Eq. (12.9), $\varepsilon^g = (z_{t_i} - \sqrt{\bar{\alpha}_{t_i}}\, D^g)/\sqrt{1 - \bar{\alpha}_{t_i}}$;

8. the guided prediction applied only in the band, $\hat{\varepsilon} = \varepsilon_{\varnothing} + (\varepsilon^g - \varepsilon_{\varnothing}) \odot m_{\text{cfg}}$.

With $\mu = 0$, no threshold and $\nu = 1$, steps 2 to 5 leave $d_i = \Delta D_i$, and by Eq. (12.10) the result is plain label guidance, Eq. (5.8); a numerical test of this identity on random inputs agrees to 4×10^{-6} in single precision. Like the other variants, EPSREC replaces only the line of the sampling loop that forms the guided prediction, the line that Experiment 18 replaced; the four branches, the regions, TCA and Local Perturbation are unchanged. Every variant regenerates the 67 resizings of the development subset.

12.3.2 Results

Table 12.5 reports the variants, and Figure 12.4 places them next to the other guidance formulations of the chapter. Taken comparison by comparison, on hard resizings:

- **Projection.** (a) against Experiment 18: 17.39 against 17.30, no difference within resolution.
- **Statistics.** (b) against (a): 17.31 against 17.39, no difference.
- **Momentum rule.** (c) against (a): 16.77 against 17.39, a reduction of 0.62 pixels, the largest single change.
- **Threshold.** (d) against (c): identical to every decimal on every group.

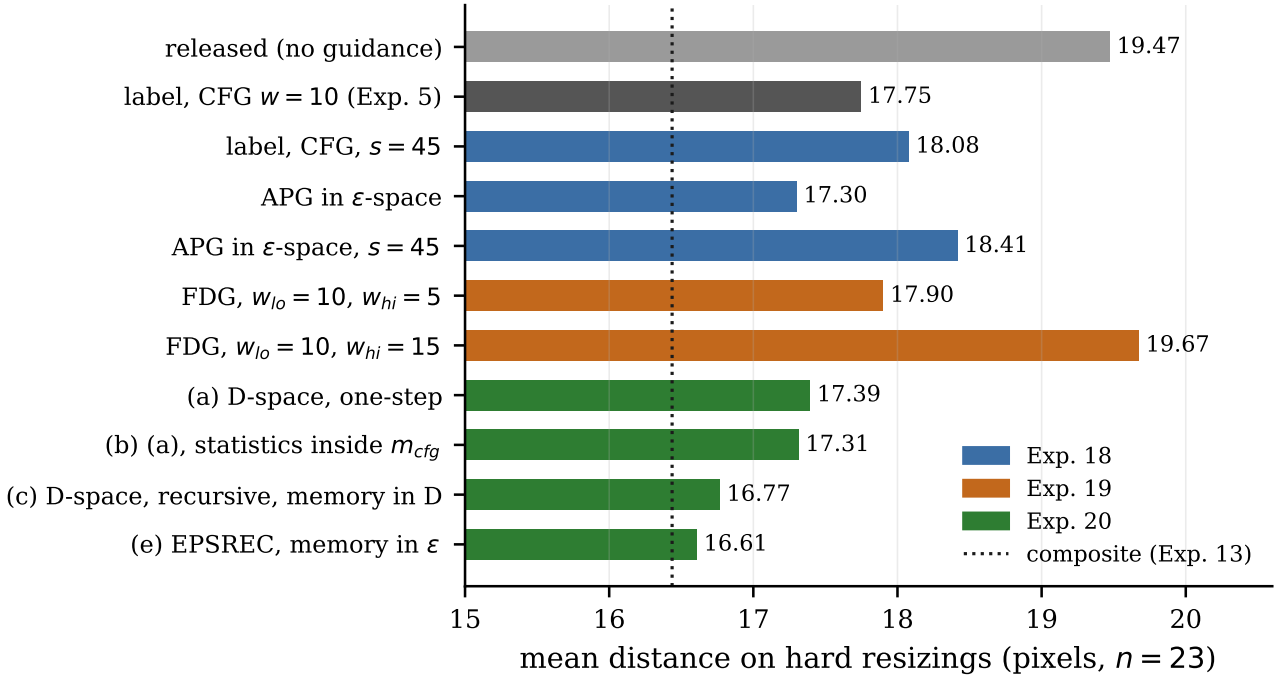


Figure 12.4: Mean distance on the 23 hard resizings for every guidance formulation of Experiments 18 to 20. The dotted line is the coarse composite scored as an output (Experiment 13).

- **Memory space.** (e) against (c): 16.61 against 16.77, a further 0.16 pixels.

EPSREC ends 1.14 pixels below label guidance and 2.87 pixels, or 14.7%, below the released pipeline. On all resizings it reaches 10.92, where label guidance had raised mean distance to 11.28. On the easy and medium resizings every form of guidance is worse than none, by 0.20 to 0.30 pixels; applied to the hard resizings only, EPSREC lowers mean distance on all resizings to 10.81. Subject consistency, background consistency and the warping error are the same for all variants.

12.3.3 Outcome

The threshold never acts. Had either threshold clipped the correction at any step of any edit, (c) and (d) would differ. They do not, so the correction stays below $5c_i$ at every step, below 0.21 at the last one. The threshold is APG's safeguard for scales far higher than $w = 10$; here it plays no role.

The projection does not matter. Removing the part along the predicted image, which changes contrast and saturation, does not move the object's geometry, and removing it along ε_p or along D_p, with image-wide or regional statistics, gives the same result.

The recursive momentum is what helps. Of the five changes, the recursive rule is the one that lowers mean distance most. A plausible reading follows from its two extreme cases: guidance that damps the push it has been repeating, and keeps its strength for what is new in the correction, steers

the reconstruction of the band around the object step by step instead of pushing it ever further in one direction.

Why the memory is stored in ε-space. With the memory in D-space, old corrections dominate late in refinement (Table 12.3); in ε-space every step uses the same weights. The 0.16 pixels between the two are within resolution. EPSREC is chosen because it is the consistent form and gives the lowest value, not because this last step is established on its own.

Guidance belongs to hard resizings. On easy and medium resizings the refined object is already close to its target and the band around it is narrow; guidance has little to reconstruct there and costs a fifth of a pixel or more. Restricted to hard resizings, EPSREC improves all resizings.

How strong the evidence is. Each change on its own is smaller than what mean distance resolves on 23 edits. The evidence for EPSREC is the size of its total effect, 2.87 pixels below the released pipeline and 1.14 below label guidance, the consistent decrease along the sequence of variants, and the absence of any cost on the other metrics. On the development subset its value on hard resizings, 16.61, is within 0.17 pixels of the composite itself. These are estimates from 23 edits, for a formulation chosen on the same edits; how much of the reduction holds on all severe resizings is measured in Chapter 15. EPSREC, with $w = 10$, $\nu = 0$, $\mu = -0.75$ and $r = 5$, is the guidance used from here on. Where it should be applied is the question of the next experiment.

12.4 Experiment 21: Where the guidance should act

12.4.1 Change

Experiments 18 to 20 guided resizings only. This experiment regenerates all 200 edits with guidance at five scopes:

- label and CFG at $w = 10$ on every edit;
- label and EPSREC on every edit;
- the routed model R1 of Experiment 14, with EPSREC in place of CFG on the resizings (latent preservation on translations, the released pipeline on rotations);
- label and EPSREC on hard resizings only, the released pipeline elsewhere, assembled from the images already generated;
- EPSREC on every edit without the label, that is, with the released empty prompt, at the settings of Experiment 20 and with $\mu = -0.5$, $\mu = 0$ and $\nu = 0.25$.

The last group tests what the algebra predicts. Without the label $\varepsilon_p = \varepsilon_\varnothing$ (Eq. (9.2)), so the correction $\Delta\varepsilon_i$ is zero at every step. In the eight steps of EPSREC (Experiment 20) the memory u_i and the

Table 12.6: Guidance applied at different scopes, all 200 edits regenerated.

	Hard rotations		Hard resizings		Easy, medium	Translations	
	SUBC↑	MD↓	SUBC↑	MD↓	res. MD↓	SUBC↑	MD↓
Released	0.8523	14.01	0.8391	19.47	7.58	0.9590	3.41
Label + CFG, all edits	0.8529	14.69	0.8391	17.75	7.80	0.9587	3.29
Label + EPSREC, all edits	0.8528	14.48	0.8391	16.61	7.78	0.9588	3.35
R1 with EPSREC	0.8523	14.01	0.8391	16.61	7.78	0.9610	3.52
Label + EPSREC, hard res. only	0.8523	14.01	0.8391	16.61	7.58	0.9590	3.41
EPSREC, no label, all edits	0.8523	14.01	0.8391	19.47	7.58	0.9590	3.41

	Whole subset ($n = 200$)					
	SUBC↑	BGC↑	MD↓	FID↓	FID_{dino}↓	KD↓
Released	0.9154	0.9657	7.60	132.28	1636.82	0.1268
Label + CFG, all edits	0.9154	0.9657	7.72	132.51	1637.89	0.1254
Label + EPSREC, all edits	0.9154	0.9655	7.56	132.30	1637.65	0.1259
R1 with EPSREC	0.9161	0.9650	7.62	132.42	1637.13	0.1313
Label + EPSREC, hard res. only	0.9154	0.9656	7.52	132.14	1637.17	0.1266
EPSREC, no label, all edits	0.9154	0.9656	7.60	132.28	1636.82	0.1268

Group sizes: hard rotations 22, hard resizings 23, easy and medium resizings 44, translations 67. Kernel distance is seeded. The four variants without the label are identical to one another on every group and metric, and differ from the released pipeline by at most 0.002 pixels in mean distance, 0.004 in FID, 0.005 in the DINOv2 distance and 0.0001 in background consistency.

correction d_i then stay zero, $D^g = D_p$, and the conversion back gives $\varepsilon^g = \varepsilon_p = \varepsilon_\varnothing$. EPSREC reduces to the unguided prediction for every value of μ, ν and r: momentum, projection and threshold act on the correction, and without the label there is no correction to act on.

12.4.2 Results

Table 12.6 reports the five scopes. On hard rotations the label worsens mean distance by 0.68 pixels with CFG and by 0.47 with EPSREC, and on all rotations by 0.16 with either (9.79 against 9.63). On translations it lowers mean distance by 0.06 to 0.12 pixels and subject consistency by 0.0002 to 0.0003. On easy and medium resizings every guided variant costs 0.20 to 0.22 pixels. EPSREC restricted to the hard resizings gives the lowest mean distance on the whole subset, 7.52, and the lowest FID, 132.14, and leaves every other edit exactly as the released pipeline makes it. The R1 variant carries the higher kernel distance of latent preservation on translations (0.1313), as in Experiment 16. The four variants without the label coincide with one another and reproduce the released pipeline to within floating-point rounding, which enters because the guided prediction is formed through the round trip to D-space and back (steps 3 to 7 of EPSREC).

12.4.3 Outcome

Guidance belongs to hard resizings. Every other edit type either loses or gains nothing measurable. On rotations the label steers the thin band around the object without supplying its orientation (Experiment 5), on translations its effects are far below what either metric resolves, and on milder

resizings it costs a fifth of a pixel. Restricted to hard resizings, EPSREC is the one scope that improves mean distance on the whole subset while leaving every other edit untouched. Its FID is the lowest in the table, but differences of this size in realism are within the variation of 200 edits (Experiment 14); what the table shows is that restricting the guidance costs no realism.

The label carries the guidance. Without the label EPSREC is inert, for every setting of the momentum and of the parallel weight, as the algebra requires. Every effect of EPSREC therefore passes through the correction that the label creates, and its settings can be compared only in the label's presence, which Experiment 20 did for the momentum. The parallel weight stays at $\nu = 0$, the value recommended by Sadat et al.

What the chapter establishes. Hypothesis 5 holds for severe resizings, and only for them. Guidance in the form of EPSREC lowers mean distance on the hard resizings of the development subset from 19.47 to 16.61 pixels, by 14.7% and to within 0.17 pixels of the composite itself, while subject consistency, the warping error and realism stay where they were and background consistency changes by less than 0.001. The gain comes from the label and from a momentum that damps repeated pushes, not from the projection or the threshold of APG, and it belongs to hard resizings: on rotations the label is harmful, on translations it does nothing measurable, and milder resizings are better served by the interior restoration of Chapter 11. EPSREC on severe resizings is the reconstruction component carried forward; on all 759 severe resizings of the benchmark it lowers mean distance by 0.34 pixels, or 1.8% (Chapter 15). Guidance acts, however, only on the band around the object; the object itself lies outside every guided region (Eq. (5.9)), and inside it refinement is left to TCA, which transplants source features at the source scale. Chapter 13 asks whether the composite's structure can be supplied there.

Chapter 13

Hypothesis 6: Supplying the Composite's Structure inside the Object

Chapter 12 steered the reconstruction of severe resizings through guidance, and EPSREC lowered mean distance on hard resizings from 19.47 to 16.61 pixels. Guidance, however, acts only on the band around the object: the object itself lies outside every guided region (Eq. (5.9)). Inside the object refinement is left to TCA, which retrieves source features by content and at the source scale (Chapter 10). For a severe resizing the coarse composite already shows the object at its target scale and position: resampled, with detail lost under enlargement and aliased under reduction (Experiment 7), but with its structure where the edit puts it. This chapter asks whether that structure can be given to the refinement inside the object.

> **Hypothesis 6.** Supplying the structure of the coarse composite inside the object, as selected frequency bands of the U-Net's own decoder features, improves the reconstruction of severe resizings beyond what guidance achieves.

Target and criterion. The target and the criterion are those of Chapter 12: mean distance on the 23 hard resizings of the development subset, where a change counts if it lowers mean distance clearly more than the one to two pixels that separate near-identical variants (Experiment 3), or consistently across related variants, without lowering subject consistency, background consistency or realism. Fusion added to guidance is compared with label guidance (17.75 pixels) and with EPSREC (16.61); fusion on its own is compared with the released pipeline (19.47).

Where the change acts. Figure 12.1 shows one refined step with both points at which the reconstruction can be steered. Guidance, the subject of Chapter 12, combines the two noise predictions in the band around the object. Spectral fusion, the subject of this chapter, corrects the U-Net's decoder features inside the object. Experiment 22 introduces the mechanism and a first sweep of its strength and band; Experiment 23 generalises it to more bands, a different reference, every edit type, with and without label guidance, and in combination with EPSREC.

Subsets. Experiment 22 regenerates only the 67 resizings of the development subset, on top of label guidance, and the 66 rotations for one variant; it reports subject consistency, background consistency, the warping error and mean distance on all resizings, on the hard resizings and on the rotations. Experiment 23 regenerates all 200 edits and reports every metric, with the kernel distance seeded.

13.1 Experiment 22: Spectral feature fusion

13.1.1 Change

Spectral fusion changes one intermediate feature map of the U-Net's decoder, inside the object. The change is explained in five steps: which features are changed (1), where the reference comes from (2), what a frequency band of a feature map is (3), the fusion rule (4) and when it acts (5). The bands and the variants of this experiment follow.

1. The features that are changed

The decoder of the U-Net rebuilds the image at increasing resolution (Table 3.1). After its second level it holds a feature map Φ with 1280 channels on a 32×32 grid; each grid position covers 16×16 image pixels. Channel by channel it is written $\Phi_k(x, y)$, with $k = 1, \ldots, 1280$ and $x, y = 0, \ldots, 31$. This map is the input of the two decoder levels that TCA modulates (blocks 10 to 15), so a change here reaches every transplanted layer. It is also the depth at which DIFT extracts its correspondence features [25]: diffusion features at this depth are known to encode where the parts of an object are. The keypoint metric uses a different network, SD-2.1, so fusion does not optimise the metric. Fusion changes Φ in the two edit rows of the batch, the unguided and the label-conditioned one; the reference branches are never touched.

2. The reference features

The inversion of §5.4.1 stores the composite's latent $z^c_{t_i}$ at every refined timestep. At refined step i the same U-Net is run on $z^c_{t_i}$, at the same timestep and with the same prompts, and stopped after the second decoder level. This partial pass, written U, gives the reference map

$$\Phi^c_i = U\big(z^c_{t_i},\, t_i\big), \tag{13.1}$$

with the same shape as Φ. It runs transformer blocks 0 to 9 only, none of which TCA modulates, so Φ^c_i describes the composite as the network sees it: the object at its target scale and position.

3. Frequency bands of a feature map

Every channel of a feature map is a 32×32 image, and like any image it can be written as a sum of waves, by the two-dimensional discrete Fourier transform:

$$\Phi_k(x, y) = \sum_{u=-16}^{15} \sum_{v=-16}^{15} \hat{\Phi}_k(u, v)\, e^{2\pi \mathrm{i}\,(ux+vy)/32}, \qquad \hat{\Phi}_k = \mathcal{F}(\Phi_k). \tag{13.2}$$

The coefficient $\hat{\Phi}_k(u, v)$ is the amount of the wave with frequency (u, v) in channel k. That wave repeats $\rho = \sqrt{u^2 + v^2}$ times across the map, so its wavelength is $32/\rho$ grid positions, or $512/\rho$ image pixels.

- The frequency $(0, 0)$ is the mean of the channel.
- Small ρ are variations that extend over large parts of the image: the broad layout. At $\rho = 3$ the wavelength is 171 pixels.
- Large ρ are variations over a few grid positions: parts, contours and texture. The finest, $\rho = 16$, have a wavelength of 32 pixels.

A *band* is chosen by a weight $\omega(u, v) \in [0, 1]$ for every frequency. The band filter keeps each frequency of each channel in proportion to its weight:

$$\mathcal{B}_\omega(\Phi) = \mathcal{F}^{-1}\big[\omega \odot \mathcal{F}(\Phi)\big]. \tag{13.3}$$

With $\omega \equiv 1$ the filter returns Φ unchanged; with $\omega \equiv 0$ it returns zero.

4. The fusion rule

At refined step i the edit features are corrected inside the object by

$$\Phi \;\leftarrow\; \Phi + \kappa_i \, M_t \odot \big(\mathcal{B}_\omega(\Phi_i^c) - \mathcal{B}_\omega(\Phi)\big), \tag{13.4}$$

where M_t is the target mask resized to 32×32 (bilinearly, so it takes values between 0 and 1 along the outline) and $\kappa_i \geq 0$ is the strength at step i.

What the rule does is clearest without the mask ($M_t \equiv 1$). The filter is linear, so in the Fourier domain the rule acts on every frequency separately:

$$\hat{\Phi}(u, v) \;\leftarrow\; \big(1 - \kappa_i \, \omega(u, v)\big)\, \hat{\Phi}(u, v) + \kappa_i \, \omega(u, v)\, \hat{\Phi}_i^c(u, v). \tag{13.5}$$

Each frequency is blended between the edit's features and the composite's, with its own weight $\kappa_i \, \omega(u, v)$. A frequency with $\omega = 0$ stays the edit's; a frequency with $\omega = 1$ moves a fraction κ_i of the way towards the composite's. The mask then confines the change to positions inside the object; outside the object Φ is left as it is.

5. When it acts

The strength is a smooth pulse at the beginning of the refined window. With the progress $p_i = (i - s)/(49 - s) \in [0, 1]$ through the fifteen refined steps,

$$\kappa_i = \kappa_0 \cdot \tfrac{1}{2}\big[1 + \cos(\pi p_i / h)\big] \quad \text{for } p_i \leq h, \qquad \kappa_i = 0 \text{ otherwise.} \tag{13.6}$$

κ_i starts at κ_0 at the first refined step and falls to zero at the fraction h of the window. For $\kappa_0 = 0.30$ and $h = 0.55$ its values at the fifteen refined steps are 0.30, 0.29, 0.25, 0.20, 0.14, 0.08, 0.03, 0.01 and then zero: fusion acts in the first eight steps, $t = 281$ to $t = 141$, while the layout of the image is formed, and leaves the last seven steps to refinement alone.

Table 13.1: Spectral feature fusion, Eq. (13.4): resizings on top of label guidance, rotations on top of the released pipeline.

	All resizings ($n = 67$)	Hard resizings ($n = 23$)		
	MD↓	SUBC↑	WRAP_E↓	MD↓
Label guidance only	11.28	0.8391	0.0587	17.75
High band, $\kappa_0 = 0.15$	11.28	0.8392	0.0587	17.80
High band, $\kappa_0 = 0.30$	11.27	0.8392	0.0587	17.03
High band, $\kappa_0 = 0.50$	11.49	0.8392	0.0586	18.01
Full band, $\kappa_0 = 0.30$	11.26	0.8393	0.0587	17.27
High band, $\kappa_0 = 0.30$, $h = 0.35$	11.28	0.8392	0.0587	17.35

	All rotations ($n = 66$)		Hard rotations ($n = 22$)	
	SUBC↑	MD↓	SUBC↑	MD↓
Released	0.8962	9.63	0.8523	14.01
High band, $\kappa_0 = 0.30$, $h = 0.45$	0.8962	9.60	0.8524	13.96

Background consistency on hard resizings: 0.9625–0.9627; on rotations background consistency and the warping error are unchanged.

The two bands. Experiment 22 uses two weights.

- **High band:** $\omega(u, v) = 1$ if $\rho > 3$ and 0 otherwise. It leaves the 29 lowest of the 1024 frequencies (the mean of each channel and every variation longer than 171 pixels) to the edit, and blends everything finer: parts, contours and texture.

- **Full band:** $\omega \equiv 1$. Every frequency is blended, and Eq. (13.4) becomes a plain blend towards the composite's features, $\Phi \leftarrow (1 - \kappa_i M_t) \odot \Phi + \kappa_i M_t \odot \Phi_i^c$.

The high band is expected to work better. The composite's lowest frequencies include the overall appearance of the inserted object and of its seam, which refinement is meant to change. The high band transfers the structure and leaves refinement free to harmonise the appearance.

Variants. On the 67 resizings, on top of label guidance at $w = 10$: the high band with $\kappa_0 \in \{0.15, 0.30, 0.50\}$ and $h = 0.55$; the full band with $\kappa_0 = 0.30$; and the high band with $\kappa_0 = 0.30$ and the shorter horizon $h = 0.35$, which acts in the first five steps. On the 66 rotations, on top of the released pipeline: the high band with $\kappa_0 = 0.30$ and $h = 0.45$, seven steps.

13.1.2 Results

At $\kappa_0 = 0.30$ the high band lowers mean distance on hard resizings to 17.03, 0.72 pixels below label guidance, and the full band to 17.27 (Table 13.1). Half the strength does nothing (17.80) and a higher one worsens the result (18.01); the shorter horizon keeps about half of the reduction (17.35). Subject consistency and the warping error stay within 0.0002 of label guidance. On rotations mean distance falls by 0.05 pixels on hard rotations and 0.03 on all rotations.

13.1.3 Outcome

A narrow effect in the expected direction. The high band helps more than the full band, as the choice of band predicted, and fusion acts only where the composite has something to give: on rotations, whose composite is itself a rotated resampling (Experiment 17), it changes nothing. Its effect on the object's pixels is small (subject consistency and the warping error, both measured inside the object, do not move), and its effect on geometry appears at one strength only and reverses at a higher one. The best reduction, 0.72 pixels, is below what mean distance resolves on 23 edits.

Next. The experiment motivates fusion but does not establish it. The next experiment tests it broadly: more bands, a different reference, every edit type, with and without label guidance, on all 200 edits.

13.2 Experiment 23: The spectral search

13.2.1 Change

Bands. Experiment 23 compares five weights $\omega(u, v)$. The first two are those of Experiment 22 and the other three are new. Writing $\rho = \sqrt{u^2 + v^2}$ for the frequency and ϕ for the direction of the vector (u, v):

- **Full:** $\omega = 1$ for every frequency.
- **High:** $\omega = 1$ for $\rho > 3$, and 0 otherwise.
- **Mid:** $\omega = 1$ for $1 < \rho \leq 3$, and 0 otherwise. Only 24 frequencies, with wavelengths between 171 and 512 pixels, are blended: the broad layout of the object at the scale of its whole extent, without the mean and without any finer detail.
- **Mid + high:** $\omega = 1$ for $1 < \rho \leq 3$ and $\omega = 0.65$ for $\rho > 3$. The layout is blended at full weight and the detail at a reduced weight.
- **Directional high:** $\omega = 0.35 + 0.65\,|\cos(\phi - \theta)|$ for $\rho > 3$, and 0 otherwise, where $\theta = r_z$ is the rotation angle of the edit. A wave whose frequency points along θ gets the weight 1 and one perpendicular to it 0.35. For translations and resizings $\theta = 0$.

Every band except the full one leaves the mean of each channel to the edit.

Reference timing. In Experiment 22 the reference at step i is $\Phi_i^c = U(z_{t_i}^c, t_i)$, Eq. (13.1): it has the same noise level as the features it corrects. A *fixed reference* uses one map at every step at which fusion acts,

$$\Phi^c = U\big(z_{181}^c,\, 181\big), \tag{13.7}$$

Table 13.2: Spectral fusion with and without label guidance, all 200 edits.

	Hard resizings		Easy, medium resizings		Hard rotations
Band	with	without	with	without	
none (reference)	17.75	19.47	7.80	7.58	14.01
high	17.03	19.91	7.76	7.59	13.91
full	17.27	22.54	7.73	7.59	14.08
mid	17.44	19.62	7.85	7.50	14.00
mid + high	16.84	22.08	7.82	7.53	13.88
directional high	17.37	20.11	7.80	7.60	13.96
high, fixed ref.	17.06	19.77	7.82	7.58	14.00
mid + high, fixed ref.	17.38	22.04	7.76	7.61	14.11

All values are mean distance in pixels. "With" and "without": on top of R1 (label guidance on resizings) or of the released pipeline. Hard rotations are the same in both, since R1 leaves rotations to the released pipeline. For every configuration, subject consistency is 0.8391–0.8393 on hard resizings and 0.8523–0.8524 on hard rotations, and mean distance on translations without the label is 3.40–3.42 (released 3.41). On the whole subset FID is 132.49–132.51 with the label (R1 alone 132.50) and 132.26–132.28 without (released 132.28); the kernel distance is 0.1311–0.1312 with the label (R1 alone 0.1312) and 0.1268 without (released 0.1268). Group sizes: hard resizings 23, easy and medium resizings 44, hard rotations 22. Kernel distance is seeded.

computed at the timestep $t = 181$, 35% of the way through the refined window. It tests whether the reference must share the noise level of the features it corrects.

With and without label guidance. Every configuration was run on top of R1 (Experiment 14), which applies label guidance at $w = 10$ to resizings, latent preservation to translations and the released pipeline to rotations, and on top of the released pipeline alone. Fusion acts on every edit type, with $\kappa_0 = 0.30$ and $h = 0.55$.

With EPSREC. The high and mid + high bands were also combined with EPSREC (Experiment 20): with the label and EPSREC on every edit, the high band within R1 with EPSREC on resizings, and both bands with EPSREC but without the label.

13.2.2 Results

With label guidance every band lowers mean distance on hard resizings below label guidance alone, to between 16.84 (mid + high) and 17.44 (mid); without it every band raises it above the released pipeline, to between 19.62 (mid) and 22.54 (full) (Table 13.2, Figure 13.1). The bands that transfer the mid and the high frequencies together, full and mid + high, are the most harmful without guidance, by 2.6 to 3.1 pixels, while either band alone costs at most 0.6. On the easy and medium resizings, on the translations and on the hard rotations no band moves mean distance by more than 0.13 pixels, and FID and the kernel distance stay those of the model that fusion is added to. Combined with EPSREC (Table 13.3), the mid + high band gives the lowest mean distance on hard resizings measured on the development subset, 16.43, 0.18 pixels below EPSREC alone, while the high band gives 17.65, 1.04 above it. Without the label, EPSREC adds nothing to either band, as Experiment 21 predicts.

Table 13.3: Spectral fusion combined with EPSREC, all 200 edits.

	Hard resizings	Hard rotations	Whole subset ($n = 200$)		
	MD↓	MD↓	MD↓	FID↓	KD↓
Label + EPSREC, all edits	16.61	14.48	7.56	132.30	0.1259
+ high band	17.65	14.76	7.71	132.28	0.1260
+ mid + high band	16.43	14.58	7.58	132.29	0.1259
R1 with EPSREC	16.61	14.01	7.62	132.42	0.1313
+ high band	17.65	13.91	7.78	132.41	0.1314
EPSREC, no label, + high band	19.94	13.91	7.78	132.28	0.1268
EPSREC, no label, + mid + high band	22.08	13.88	7.78	132.27	0.1268

Hard resizings $n = 23$, hard rotations $n = 22$. Kernel distance is seeded.

13.2.3 Outcome

Fusion works only together with guidance. The same transfer of the composite's structure improves hard resizings when the label steers the band around the object and worsens them when it does not; transferring the object's layout and its detail together is the most harmful without the label, and mid + high the most useful with it. The two mechanisms act on complementary regions (Figure 12.1): guidance reconstructs the band outside the object towards what the object is, and fusion imposes the composite's structure at the new scale inside it. The most likely explanation is that imposing the object's structure at the new scale inside, while the band around it is regenerated from the scene alone, produces an object whose interior and surroundings disagree, and that the label removes the disagreement by reconstructing the band consistently with the object.

The choice of band matters less than the guidance. With the label, the seven configurations lie within 0.6 pixels of one another, within what mean distance resolves, and neither the fixed reference nor the directional weighting is distinguishable from the plain bands. Mid + high gives the lowest values, with CFG and with EPSREC.

Fusion does not reliably add to EPSREC. Added to EPSREC, one band lowers mean distance by 0.18 pixels and the other raises it by 1.04: effects of opposite sign, both within resolution. Once guidance takes the form of EPSREC, the contribution of fusion cannot be separated from the variation of the metric, and on rotations fusion leaves the cost of the label unchanged.

What the chapter establishes. Hypothesis 6 is not supported. It claims an improvement beyond what guidance achieves, and on top of EPSREC, the guidance established in Chapter 12, the contribution of fusion cannot be separated from the variation of the metric: the mid + high band gives 16.43 pixels against 16.61 for EPSREC alone, and the high band gives 17.65. What the experiments show instead is an interaction with guidance. Supplying the composite's structure inside the object lowers mean distance on hard resizings when label guidance steers the band around the object, from 17.75 pixels to between 16.84 and 17.44 depending on the band, and raises it when it does not, to as much as 22.54. Which band is transferred matters less than whether guidance is present. Fusion does not help rotations, and it moves neither subject consistency, the warping error nor realism. Spectral fusion is therefore not a component in its own right: EPSREC remains the reconstruction for severe resizings,

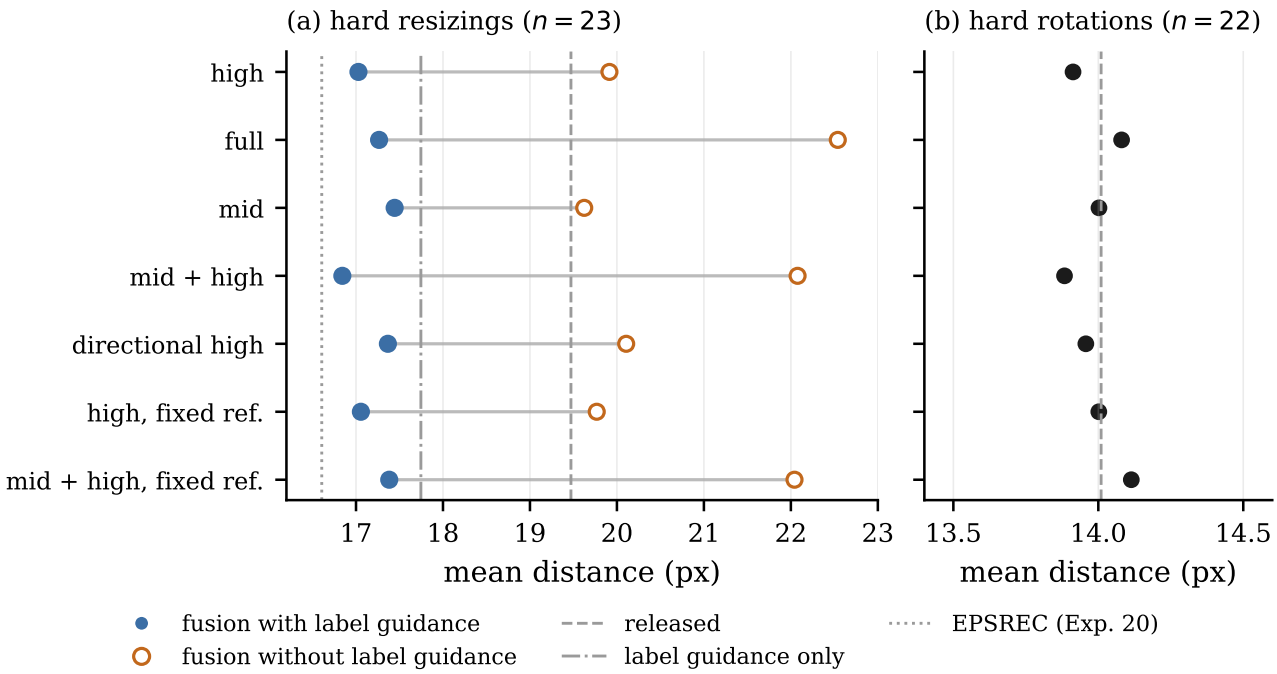


Figure 13.1: Mean distance for every band of spectral fusion. (a) Hard resizings, with label guidance (filled) and without it (open); the vertical lines mark the released pipeline, label guidance alone and EPSREC. (b) Hard rotations, against the released pipeline.

and mid + high fusion is carried forward only as an optional addition, evaluated as an ablation of the final model. Chapter 14 assembles the components of Chapters 11 and 12 into one model that chooses among them from the parameters of each edit.

Chapter 14

Selective Geometric Refinement

Chapters 11 to 13 produced components that each help one kind of edit. Restoring the object's interior from the composite after refinement helps translations and moderate resizings and harms rotations (Chapter 11). Guidance with the object label in the form of EPSREC helps severe resizings, harms rotations and costs the milder resizings (Chapter 12). Spectral fusion adds nothing reliable on top of EPSREC (Chapter 13). And no component improves rotations. Applied to every edit, each component damages the edits it does not suit (Experiments 16 and 21). The components are therefore combined by *routing*: every edit is sent to the one branch that suits it, and edits that no component helps are left to FreeFine unchanged.

This chapter turns that principle into a method that can be applied to any edit. Section 14.1 sets out the four branches and where each comes from. Experiment 24 replaces the severity tertiles used during development by a rule computed from the requested transformation alone. Section 14.3 defines the resulting method, SGR-EPSREC (Selective Geometric Refinement with EPSREC guidance). Experiment 25 assembles it on the development subset and checks that each branch behaves on the router's own groups as it did in the experiment that produced it. Chapter 15 evaluates the method on the whole benchmark.

14.1 The four branches

Table 14.1 lists the branches. Each is the component that the earlier experiments established for one kind of edit; no setting was changed for the router.

Two principles decided the choice. A branch is chosen by the metrics that measure the failure it addresses (subject consistency and mean distance) and not by the realism metrics, whose differences on 200 edits are within their variation (Experiment 14). And between components that the metrics cannot separate, the simpler one is kept: this is why spectral fusion is not part of the severe branch (Chapter 13).

Table 14.1: The four branches of the router and the experiments that established them.

Edit	Branch	Established in
Translation	FreeFine refinement with interior restoration, Eq. (11.2): the object's interior is restored from the composite, and a band of $w = 4$ pixels along the outline keeps the refined result	Experiments 15 and 17
Rotation	FreeFine, unchanged	no component helps: Experiments 8–11, 17, 21, 22–23
Resizing, moderate	FreeFine refinement with interior restoration, $w = 8$	Experiment 17
Resizing, severe	refinement guided by the object label with EPSREC ($w = 10$, $\mu = -0.75$, $\nu = 0$, $r = 5$); no interior restoration	Experiments 20 and 21

14.2 Experiment 24: A routing rule from the requested transformation

14.2.1 Change

Why the development grouping cannot route. The experiments of Chapters 9 to 13 grouped edits by the severity tertiles of Experiment 2, Eq. (7.2). That score cannot route a new edit, for two reasons. It is normalised and cut at quantiles computed over all 5,677 edits of the benchmark, so the tertile of a single edit is not defined without the whole population. And it includes the fraction of the image covered by the object, so it measures more than the transformation. A router has to decide from what a user provides: the source image, the object mask and the requested transformation.

The edit type. The type is read from the parameters as in Experiment 14: a rotation if $r_z \neq 0$, a resizing if $(s_x, s_y) \neq (1, 1)$, and a translation otherwise. Every edit of GeoBench-2D activates exactly one kind of transformation (Table 5.3), so this reading is exact.

The severity of a resizing. A resizing is described by the single scale $s = \sqrt{s_x s_y}$ (§5.6). The authors of GeoBench define the difficulty of a resizing instruction directly by its scale (Table 14.2) [34]. The rule adopts their hard ranges:

$$\text{severe}(s) \quad \Longleftrightarrow \quad s \geq 1.5 \;\; \text{or} \;\; s \leq 0.6. \tag{14.1}$$

The thresholds are fixed by the benchmark's own definition; no metric of any method was consulted to choose them.

The audit. The rule was applied to the transformation parameters of all 5,677 edits, read from the annotation that the pipeline itself uses (a fixed manifest, SHA-256 beginning `17d97110`). Two proper-

Table 14.2: Scale ranges of resizing instructions at each difficulty level, as defined by the authors of GeoBench [34].

Difficulty	Enlargement, s	Reduction, s
Easy	1.1–1.3	0.8–0.9
Medium	1.3–1.5	0.6–0.8
Hard	1.5–3.0	0.4–0.6

Table 14.3: The 2,635 resizings of GeoBench-2D divided by the routing rule, Eq. (14.1).

	Edits	Scale range
Enlargement, moderate	831	1.1001–1.4985
Enlargement, severe	236	1.5005–2.8804
Reduction, moderate	1,045	0.6001–0.9000
Reduction, severe	523	0.4001–0.5985
Moderate	1,876	
Severe	759	

ties were checked first: every resizing is isotropic, $s_x = s_y$ exactly, and every translation and rotation has $s_x = s_y = 1$ exactly. The rule is therefore defined for every edit.

14.2.2 Results

The rule classifies 759 resizings, 28.8%, as severe: 236 enlargements and 523 reductions (Table 14.3). No edit lies exactly on a threshold, so it does not matter to which side a boundary belongs. Against the severity tertiles (Table 14.4), 627 of the 800 resizings in the hard tertile are severe (78%), and 627 of the 759 severe resizings are in the hard tertile (83%); the others are 130 medium and 2 easy. On the development subset the rule gives 20 severe and 47 moderate resizings, and it classifies 9 of the 67 resizings (13%) differently from the tertile: 6 resizings of the hard tertile are moderate by scale, and 3 severe ones are not in the hard tertile.

14.2.3 Outcome

A rule that can be applied to any edit. The rule reads only the requested scale, is fixed before any result, and coincides with the benchmark authors' definition of a hard resizing.

Related to the development grouping, but not the same. Most severe resizings are in the hard tertile and most of the hard tertile is severe, but the two groupings differ, because the tertile also counts the object's area: a moderately rescaled large object can fall in the hard tertile, and a severely rescaled small one outside it. EPSREC was developed on the hard tertile, and the router applies it to a different group. Its effect has to be measured on the router's own groups before the full evaluation; Experiment 25 does so.

Table 14.4: Resizings classified by the routing rule against the severity tertile of Experiment 2, on the whole benchmark and on the development subset.

	Severity tertile	Moderate	Severe
All resizings ($n = 2{,}635$)	easy	972	2
	medium	731	130
	hard	173	627
Development subset ($n = 67$)	easy and medium	41	3
	hard	6	17

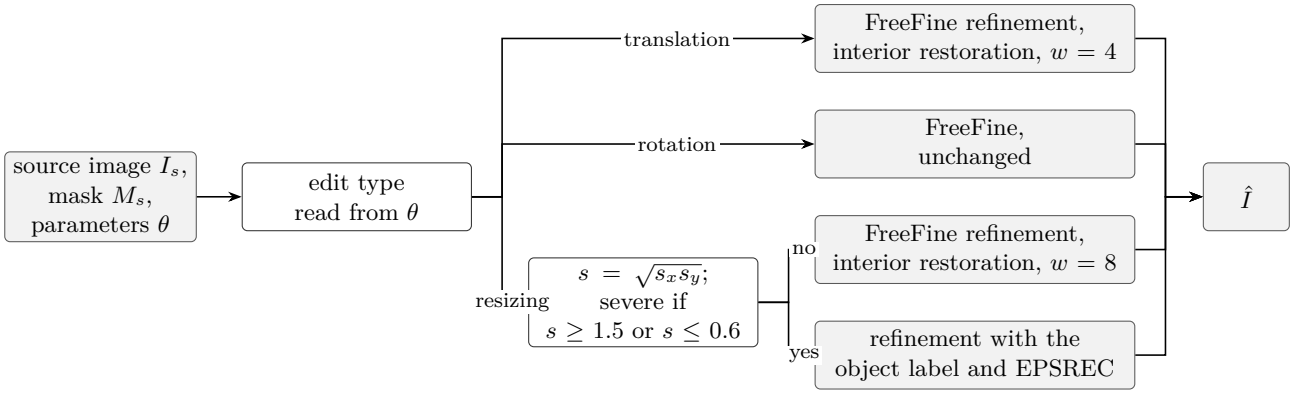


Figure 14.1: The SGR-EPSREC router. The edit type is read from the requested parameters, and a resizing is classified as severe by its scale alone, Eq. (14.1). Three branches use FreeFine's refinement, two of them followed by interior restoration, Eq. (11.2); severe resizings are refined with the object label and EPSREC guidance.

14.3 SGR-EPSREC

Given a source image I_s, an object mask M_s and the requested parameters θ, let $\hat{I}_{\mathrm{FF}}$ be the output of FreeFine as released (Chapter 5), $\mathcal{R}_w$ interior restoration, Eq. (11.2), with a refined band of width w, and $\hat{I}_{\mathrm{EPS}}$ the output of refinement with the object label as prompt and EPSREC guidance, the eight steps of §12.3. SGR-EPSREC returns

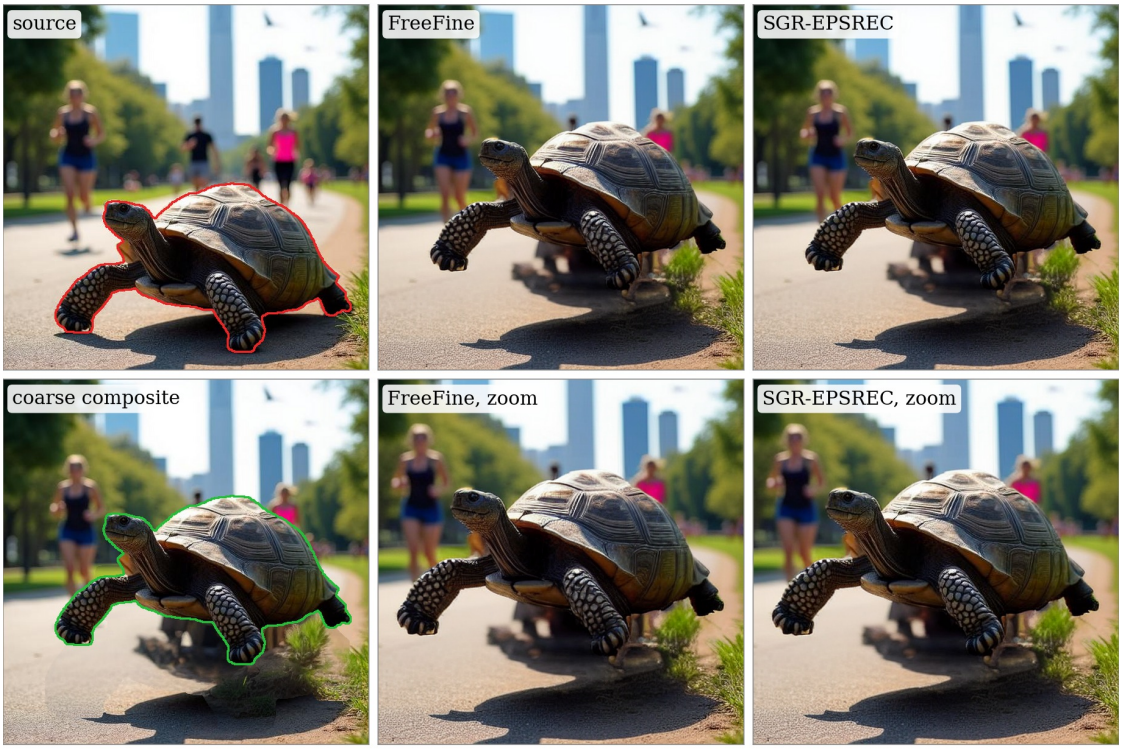
$$\hat{I} = \begin{cases} \mathcal{R}_4(\hat{I}_{\mathrm{FF}}) & \text{for a translation,} \\ \hat{I}_{\mathrm{FF}} & \text{for a rotation,} \\ \mathcal{R}_8(\hat{I}_{\mathrm{FF}}) & \text{for a resizing that is not severe,} \\ \hat{I}_{\mathrm{EPS}} & \text{for a severe resizing, Eq. (14.1).} \end{cases} \tag{14.2}$$

Figure 14.1 shows the router.

Figures 14.2 and 14.3 show one edit sent to each branch. On the translation and the moderate resizing interior restoration keeps the source's detail inside the object (the lettering on the bottle stays legible where FreeFine redraws it), and the rotation is FreeFine's own output. On the severe enlargement the two outputs differ only slightly: EPSREC acts on where the object's parts are placed rather than on how they look.

The method has five properties.

(a) Translation: 87 pixels upwards



(b) Moderate resizing: to 1.16 of the original size



Figure 14.2: FreeFine against SGR-EPSREC on the two branches that restore the object's interior: (a) a translation and (b) a moderate resizing. In each part, the top row shows the source image (red: the object's mask) and the whole outputs; the bottom row shows the coarse composite (green: the target mask) and the region in the yellow box, enlarged.

(c) Rotation: 16 degrees



(d) Severe resizing: to 2.33 of the original size

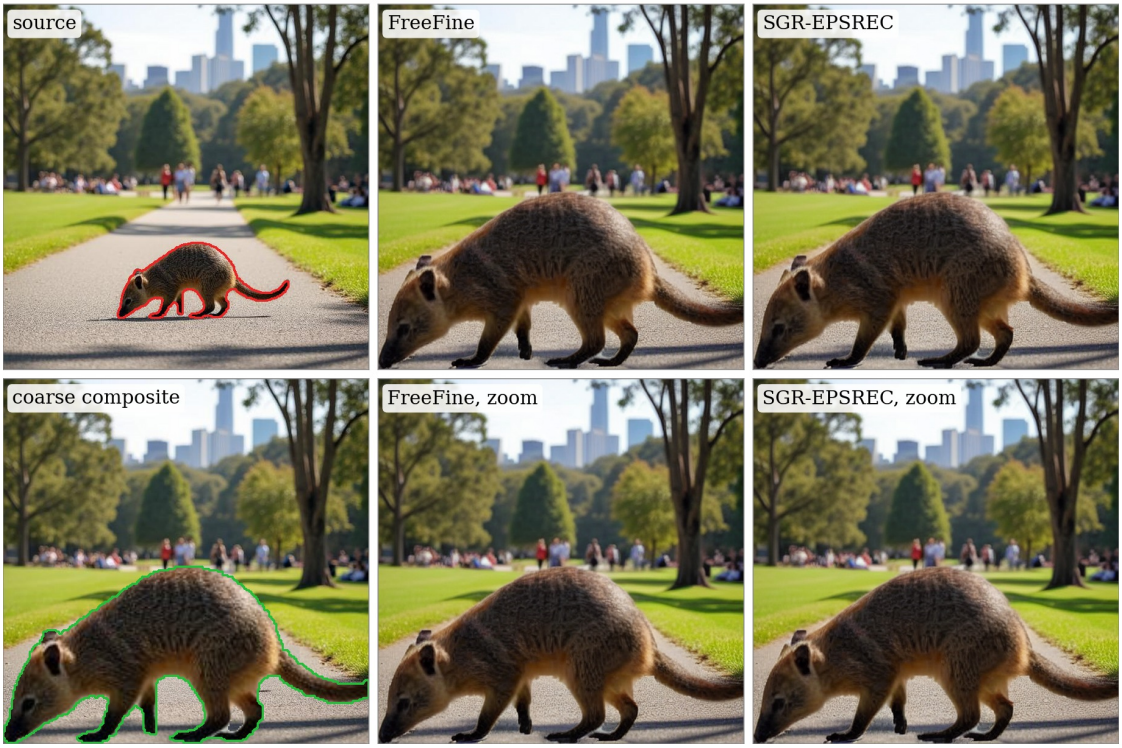


Figure 14.3: FreeFine against SGR-EPSREC on (c) a rotation, which is left to FreeFine and is identical, and (d) a severe resizing, refined with EPSREC. Laid out as in Figure 14.2.

- **Training-free.** No parameter is learned. The U-Net, the autoencoder and every setting of FreeFine are unchanged.

- **It uses only the request.** The router reads the source image, the object mask and the transformation parameters. The benchmark's difficulty labels and the severity tertiles are never read.

- **Exact fallback.** With routing disabled, every edit goes through FreeFine and the output reproduces it byte for byte (the switch-off check of §8.3). Rotations are always identical to FreeFine's output.

- **Mostly post-processing.** Three of the four branches take FreeFine's output and at most restore the object's interior in it; only severe resizings are generated differently.

- **Partly improved by construction.** Interior restoration takes the object's interior from the coarse composite distributed with the benchmark, which is also the reference of the warping error. On the restored branches the warping error, and part of subject consistency, improve by construction (rule 4 of §8.5).

14.4 Experiment 25: The router on the development subset

14.4.1 Change

SGR-EPSREC was assembled on the 200 edits of the development subset from images that had already been generated: FreeFine's outputs, with interior restoration applied to translations ($w = 4$) and moderate resizings ($w = 8$), and, for the severe resizings, the outputs of the label and EPSREC run of Experiment 21. Generation is deterministic, so these are exactly the images the router produces for these edits. A second version, SGR-MIDHF-EPSREC, adds mid + high spectral fusion ($\kappa_0 = 0.30$, $h = 0.55$) to the severe branch, the combination that gave the lowest mean distance on the hard tertile in Chapter 13, and is kept as the ablation of the method.

Both versions were scored with all seven metrics, with mean distance and the kernel distance seeded, on the whole subset, on the three edit types and on the router's own groups of resizings: 20 severe and 47 moderate. Before any variant was scored, FreeFine's outputs were scored again and reproduced the values of every earlier study (subject consistency 0.9154, background consistency 0.9657, warping error 0.0488, mean distance 7.6023).

14.4.2 Results

Tables 14.5 to 14.7 report the router. Group by group:

- **Translations:** subject consistency +0.0080, warping error −77%, mean distance +0.55 pixels, the values of Experiment 15, since the images are the same.

- **Rotations:** identical on every metric.

Table 14.5: SGR-EPSREC against FreeFine on the development subset, by edit type and by the router's groups of resizings.

		SUBC↑		WRAP_E↓		MD↓	
Group	n	FreeFine	SGR	FreeFine	SGR	FreeFine	SGR
Translations	67	0.9590	0.9670	0.0495	0.0115	3.41	3.96
Rotations	66	0.8962	0.8962	0.0423	0.0423	9.63	9.63
Resizings	67	0.8906	0.8946	0.0544	0.0312	10.93	10.11
moderate	47	0.9214	0.9270	0.0532	0.0201	7.70	7.30
severe	20	0.8185	0.8184	0.0573	0.0573	22.08	20.80
Whole subset	200	0.9154	0.9194	0.0488	0.0283	7.60	7.52

Background consistency, FreeFine against SGR: 0.9639 against 0.9644 on translations, 0.9661 against 0.9664 on severe resizings, 0.9657 against 0.9659 on the whole subset, and unchanged on the other groups. Mean distance is averaged over keypoints, so the value of a group is not the edit-weighted average of its sub-groups (§5.8).

Table 14.6: Realism of SGR-EPSREC against FreeFine on the development subset.

	FID↓		FID_{dino}↓		KD↓	
Group	FreeFine	SGR	FreeFine	SGR	FreeFine	SGR
Translations	200.04	199.99	2444.54	2449.76	0.0259	0.0213
Rotations	205.96	205.96	2502.44	2502.44	0.1344	0.1344
Resizings	201.26	201.65	2465.28	2466.72	0.1156	0.1133
Whole subset	132.28	132.46	1636.82	1637.12	0.1268	0.1256

Kernel distance is seeded.

- **Moderate resizings:** subject consistency +0.0056, warping error −62%, mean distance −0.40 pixels.
- **Severe resizings:** mean distance 22.08 → 20.80, −1.28 pixels or 5.8%; subject consistency and the warping error do not move.
- **Whole subset:** subject consistency +0.0040, warping error −42%, mean distance −0.08 pixels; FID +0.18, the DINOv2 distance +0.30 and the kernel distance −0.0012.
- **Ablation:** spectral fusion changes only the severe resizings, where mean distance is 20.98 against 20.80.

14.4.3 Outcome

Each branch behaves as in the experiment that produced it. On the router's own groups interior restoration gives moderate resizings +0.0056 in subject consistency, −62% in warping error and −0.40 pixels in mean distance, close to what Experiment 17 found on the easy and medium tertiles (+0.0061, −57%, −0.26). Translations carry their declared cost in mean distance, and rotations are untouched. The router has not changed what its components do.

Table 14.7: SGR-EPSREC against its ablation with mid + high spectral fusion in the severe branch.

	MD↓			FID↓
	Severe resizings	Resizings	Whole subset	Whole subset
FreeFine	22.08	10.93	7.60	132.28
SGR-EPSREC	20.80	10.11	7.52	132.46
SGR-MIDHF-EPSREC	20.98	10.17	7.54	132.45

The two versions are identical on every edit except the severe resizings, where their subject consistency (0.8184, 0.8185) and warping error (0.0573) are also the same.

The severe branch keeps its direction with a smaller gain. On the router's 20 severe resizings EPSREC lowers mean distance by 1.28 pixels, against 2.87 on the 23 edits of the hard tertile, from a higher starting value (22.08 against 19.47). The two groups share 17 edits, and groups of 20 and 23 edits cannot separate a real difference in the size of the gain from the variation of the metric. What holds on both is the direction.

The whole subset moves little on mean distance. The translations' cost, +0.55 pixels on 67 edits, nearly offsets the resizings' gain, and the value over the whole subset is weighted by keypoint counts, so it falls by only 0.08 pixels. The realism metrics move in opposite directions (FID up by 0.18, the kernel distance down by 0.0012), and both differences are within the variation of 200 edits (Experiment 14).

Spectral fusion is left out. On the router's severe group fusion makes mean distance worse (20.98 against 20.80), whereas on the hard tertile it made it better (16.43 against 16.61). The sign of its contribution changes with the grouping, which confirms Chapter 13: its effect is within the variation of the metric. The simpler SGR-EPSREC is the method; SGR-MIDHF-EPSREC is carried forward as its ablation.

What this experiment is. The images were assembled on the edits on which every branch was chosen. The experiment checks that the pieces fit together and that the router does not change what they do; it is not evidence that the method generalises.

What the chapter establishes. SGR-EPSREC is a training-free router over the frozen FreeFine pipeline with four branches: interior restoration for translations and moderate resizings, FreeFine unchanged for rotations, and label guidance with EPSREC for severe resizings. It reads only the requested transformation, and its one threshold is the benchmark authors' definition of a hard resizing. On the development subset every branch reproduces, on the router's own groups, the behaviour of the experiment that produced it: consistency gains on translations and moderate resizings, with a declared cost in mean distance on translations, a geometric gain on severe resizings, rotations untouched and realism within its variation. Chapter 15 evaluates the method and its ablation on all 5,677 edits of the benchmark, of which the development subset is 3.5%.

Chapter 15

Full-Scale Evaluation

Chapter 14 assembled SGR-EPSREC on the 200 edits of the development subset, the edits on which its branches were chosen. This chapter evaluates it on all 5,677 edits of GeoBench-2D (Experiment 26), the scale at which the benchmark reports its results and at which the realism metrics can be read. It shows what the method changes, on which edits, and how much of the improvement reflects better editing.

15.1 Evaluation setup

Three models were generated in the same run: FreeFine as released, SGR-EPSREC (Eq. (14.2)), and its ablation SGR-MIDHF-EPSREC, which adds mid + high spectral fusion to the severe branch (Experiment 25). FreeFine was regenerated rather than taken from Chapter 7, and the switch-off check of §8.3 confirmed that the new images are bit-identical to those of the reproduction. The branches of the router were then computed from these images: rotations are copied unchanged, translations and moderate resizings receive interior restoration (Eq. (11.2), $w = 4$ and $w = 8$), and the 759 severe resizings are generated with the object label and EPSREC. Before scoring, the rotations were checked to be byte-identical in all three models.

All seven metrics were computed with the authors' evaluation code, with mean distance and the kernel distance seeded (§8.4), on the whole benchmark and on each branch of the router: 1,439 translations, 1,603 rotations, 1,876 moderate and 759 severe resizings.

15.2 Results

On all 5,677 edits SGR-EPSREC is better than FreeFine on all seven metrics of the benchmark (Table 15.1). Divided by branch (Table 15.2), the change has the pattern that the development subset predicted (Experiment 25).

- **Translations:** subject consistency +0.0073 and warping error −75%, paid for with +0.19 pixels (+5%) of mean distance.

- **Rotations:** identical to FreeFine on every metric.

Table 15.1: FreeFine, SGR-EPSREC and its ablation on all 5,677 edits of GeoBench-2D.

Metric	FreeFine	SGR-EPSREC	SGR-MIDHF-EPSREC	Change	Relative
FID↓	35.0439	34.7793	34.7788	−0.2646	−0.8%
FID_{dino}↓	487.8231	487.3580	487.3660	−0.4651	−0.1%
KD↓	0.1433	0.1417	0.1417	−0.0016	−1.1%
SUBC↑	0.9113	0.9144	0.9144	+0.0031	+0.3%
BGC↑	0.9670	0.9671	0.9671	+0.0001	+0.0%
WRAP_E↓	0.0474	0.0292	0.0292	−0.0182	−38.4%
MD↓	8.5024	8.1779	8.1825	−0.3245	−3.8%

Change and relative change are SGR-EPSREC against FreeFine. Mean distance and the kernel distance are seeded.

Table 15.2: FreeFine against SGR-EPSREC on each branch of the router.

		SUBC↑		WRAP_E↓		MD↓	
Branch	n	FreeFine	SGR	FreeFine	SGR	FreeFine	SGR
Translations	1,439	0.9592	0.9665	0.0489	0.0121	3.80	3.99
Rotations	1,603	0.9008	0.9008	0.0433	0.0433	10.47	10.47
Moderate resizings	1,876	0.9290	0.9325	0.0453	0.0187	7.24	7.20
Severe resizings	759	0.7991	0.7995	0.0581	0.0581	19.04	18.70

Background consistency changes by at most 0.0002 on any branch.

- **Moderate resizings:** subject consistency +0.0035, warping error −59%, mean distance essentially unchanged (−0.04 pixels).

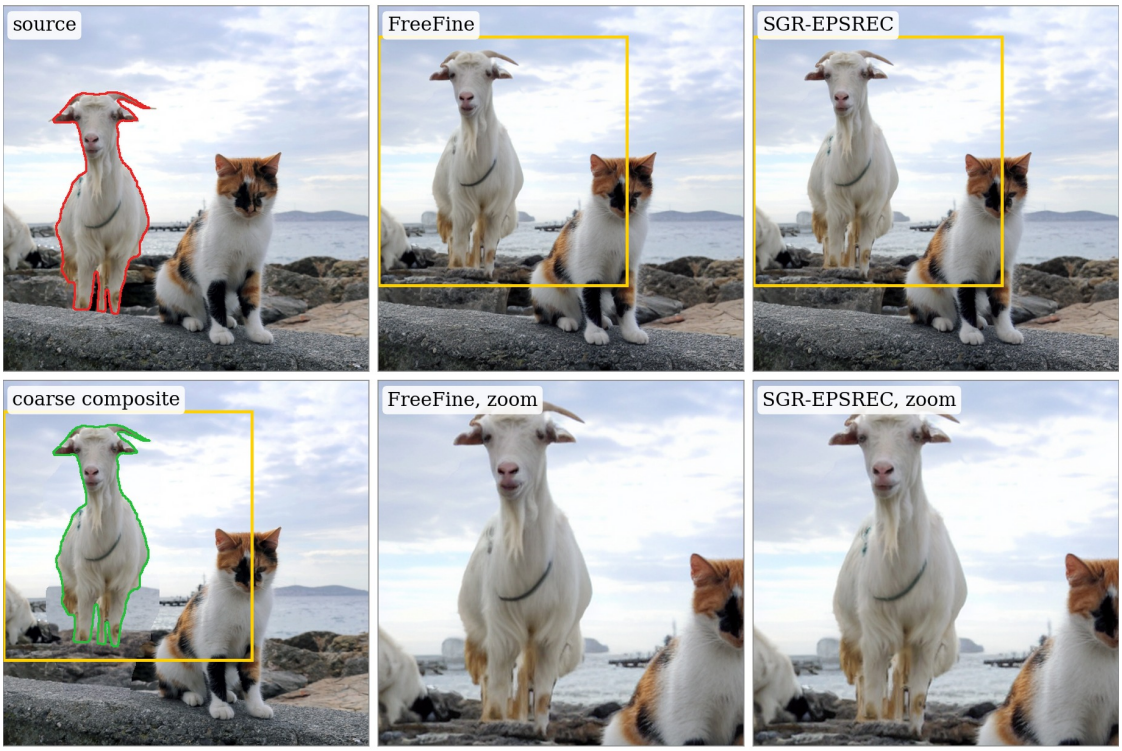
- **Severe resizings:** mean distance −0.34 pixels (−1.8%), subject consistency +0.0004.

The gain on severe resizings is smaller than on the development subset, where it was 1.28 pixels on 20 edits. This is expected: EPSREC was chosen as the best of several guidance formulations on a few edits (Chapter 12), and a value selected that way overstates the effect; on 759 edits the measurement is far more reliable.

Realism is not lowered. FID, the DINOv2 distance and the kernel distance are all lower than FreeFine's on the whole benchmark, although interior restoration inserts the composite's interior into more than 3,300 edits and the composite alone scored a worse realism than refinement (Experiment 13). The differences are small, smaller than the difference between the reproduced and the published FreeFine (Table 7.3), so they show that realism is preserved rather than improved.

Figures 15.1 and 15.2 show FreeFine and SGR-EPSREC on edits of the full benchmark: typical edits of the two branches that restore the interior, whose change in subject consistency is the median of their branch, and the edits on which subject consistency fell the most among moderate and severe resizings. On typical edits the two outputs are nearly indistinguishable and differ in fine detail. Even the largest losses are not visible failures: the outputs differ in texture and sharpness, not in shape or placement.

(a) Translation: 59 pixels upwards

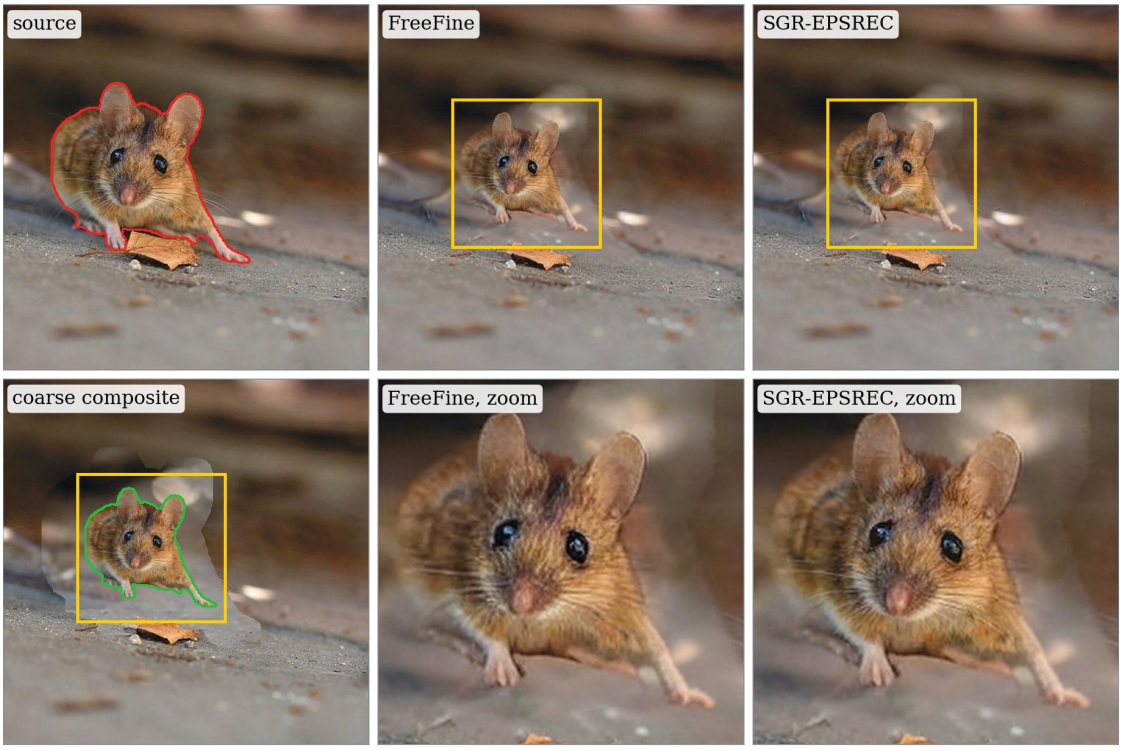


(b) Moderate resizing: to 1.34 of the original size



Figure 15.1: FreeFine against SGR-EPSREC on typical edits of the full benchmark, whose change in subject consistency is the median of their branch: (a) a translation and (b) a moderate resizing. In each part, the top row shows the source image (red: the object's mask) and the whole outputs; the bottom row shows the coarse composite (green: the target mask) and the region in the yellow box, enlarged. The edits are chosen among those outside the development subset whose object covers 6 to 30% of the image.

(a) Moderate resizing: to 0.66 of the original size



(b) Severe resizing: to 0.46 of the original size



Figure 15.2: The edits with the largest loss in subject consistency on (a) moderate resizings (−0.050) and (b) severe resizings (−0.035), chosen as in Figure 15.1 and laid out in the same way.

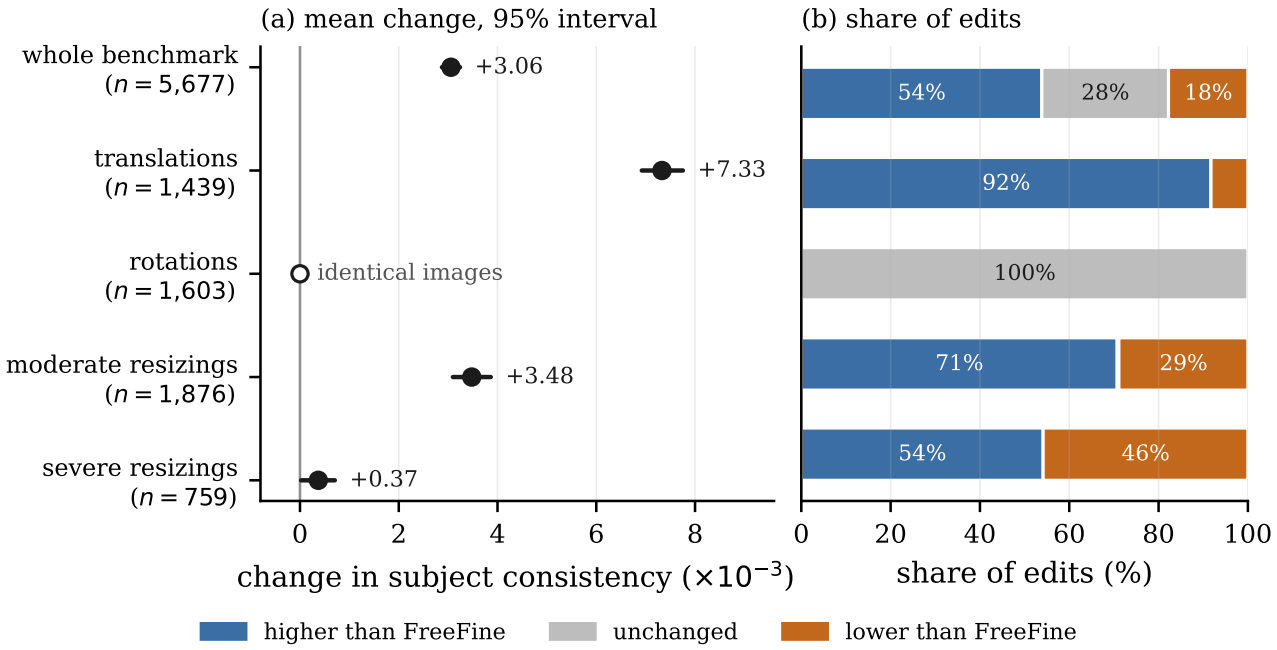


Figure 15.3: Change in subject consistency edit by edit, SGR-EPSREC minus FreeFine, by branch. (a) Average change with its 95% interval. (b) Share of edits whose subject consistency is higher, unchanged or lower.

15.3 How much of the improvement is real

Seven better numbers do not all carry the same weight. Four questions separate what the method achieves from what follows from how it, or the metrics, are built.

Is the gain in subject consistency more than noise? Yes. Subject consistency is computed for every edit, and each edit is generated by both models from the same inputs, so the two values of an edit can be subtracted and the difference shows the effect of the method on that edit alone. Figure 15.3 summarises these differences. Over the whole benchmark the average rise is 0.0031, and a bootstrap over the edits places it between 0.0029 and 0.0032 with 95% confidence, far from zero. The rise is not the work of a few edits: 54% of the edits improve, 18% get worse, and the rest are the rotations, which do not change. It is strongest on translations (92% of them improve) and moderate resizings (71%). On severe resizings it is real but small: 54% improve and 46% get worse, and a Wilcoxon signed-rank test gives $p = 0.043$, just below the usual 0.05 threshold.

Which gains are guaranteed by the method? The warping error measures how far the output is from the coarse composite, and interior restoration copies the composite into the object's interior. The 38% reduction is therefore guaranteed on the restored branches, and it is absent on severe resizings, which are generated anew; it is reported but not counted as evidence. For the same reason part of the gain in subject consistency on the restored branches is expected, since the restored interior is the source object itself; on severe resizings nothing is copied, and there the gain is the small one above. The rise of 0.0001 in background consistency is also a side effect: the released metric code counts a thin line of object pixels along the source outline as background, and restoration changes some of them.

What does the −3.8% in mean distance mean? Mean distance is averaged over keypoints, not over edits, and each model is scored on the keypoints it produces (§5.8). SGR-EPSREC changes how many keypoints each kind of edit contributes: the values of Table 15.2 imply that in its resizings the severe ones supply 20.5% of the keypoints, against 23.1% in FreeFine's. Because severe resizings have a large mean distance, giving them less weight lowers the average even where no edit has improved. Held at FreeFine's weights, the whole-benchmark change would lie between −0.01 and +0.01 pixels: the cost on translations and the gain on severe resizings cancel. The −3.8% is the official value of the metric and is reported as such, but the geometric result of the method is the change within each branch: −0.34 pixels on severe resizings and +0.19 on translations.

Does the result depend on the development edits or on the ablation? No. The branches were chosen on the 200 edits of the development subset, which belong to the benchmark; without them, the per-edit gain in subject consistency is 0.0030 instead of 0.0031, and every branch keeps its value to within 0.0001. The ablation, SGR-MIDHF-EPSREC, differs only on severe resizings, where spectral fusion makes mean distance worse (18.78 against 18.70 pixels) and changes nothing else appreciably. SGR-EPSREC, the simpler of the two, remains the method.

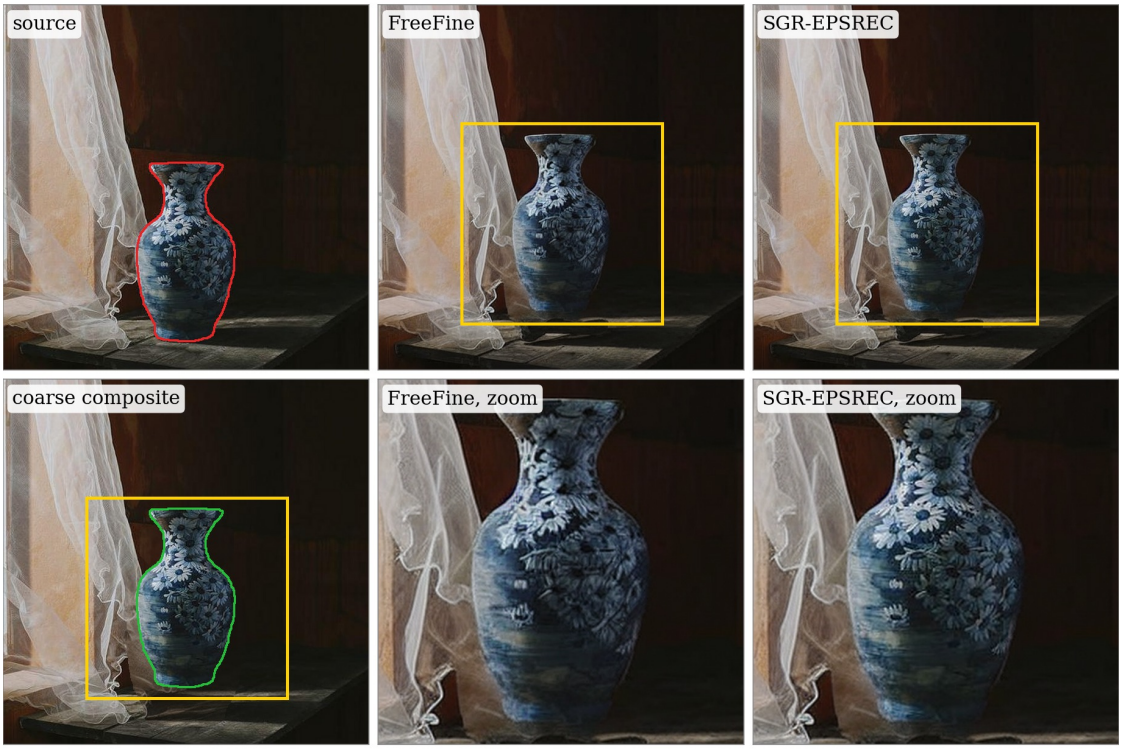
15.4 A gallery of edits

Figures 15.4 to 15.6 show six more edits of the full benchmark, four translations and two moderate resizings, chosen because the effect of interior restoration can be seen with the eye. They are not a random sample: their gains in subject consistency range from +0.0027 to +0.0164, while typical edits are those of Figure 15.1. All six show the same pattern. FreeFine's refinement smooths or rewrites the fine structure inside the object: the painted flowers of the vase, the screen of the toy, the woven surface of the mat, the lettering on the bottle, the flowers of the headdress and the contents of the case. SGR-EPSREC keeps that structure as it is in the source, because the interior of its output is the source object, placed by the transformation; around the object the two outputs coincide.

15.5 Summary

On the whole benchmark SGR-EPSREC improves all seven metrics of GeoBench-2D over FreeFine, leaves rotations exactly as FreeFine produces them, and does not lower realism. The improvement in subject consistency is solid, comes from most edits, and is largest where interior restoration acts. The geometric gain is concentrated on severe resizings (−0.34 pixels), and translations pay 0.19 pixels of mean distance for their gain in consistency. The warping error, background consistency and the whole-benchmark mean distance improve partly or wholly because of how the method and the metrics are built, and are not counted as evidence. Chapter 16 relates these results to the hypotheses of the thesis.

(a) Translation: 40 pixels upwards

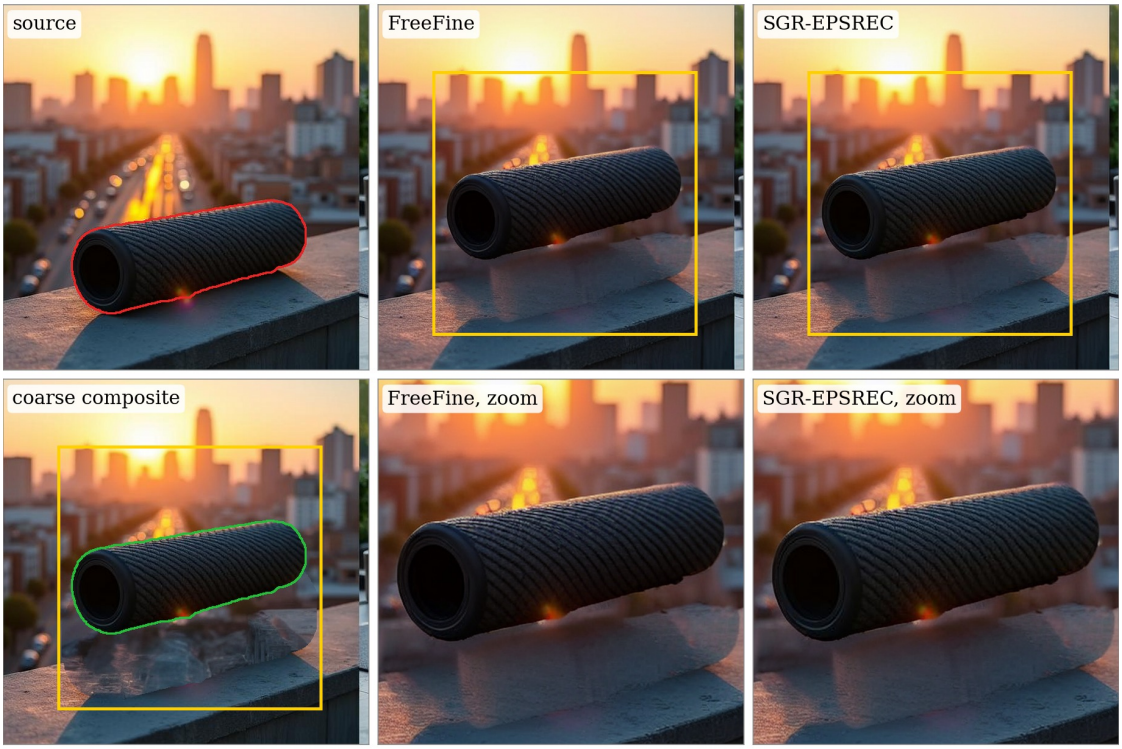


(b) Translation: 110 pixels to the left



Figure 15.4: Two translations of the full benchmark on which interior restoration is visible. Change in subject consistency, SGR-EPSREC against FreeFine: (a) +0.0164, (b) +0.0058. Layout as in Figure 15.1.

(c) Translation: 76 pixels upwards



(d) Translation: 143 pixels to the left



Figure 15.5: Two more translations, chosen and laid out as in Figure 15.4. Change in subject consistency: (c) +0.0120, (d) +0.0027.

(e) Moderate resizing: to 0.87 of the original size



(f) Moderate resizing: to 0.64 of the original size

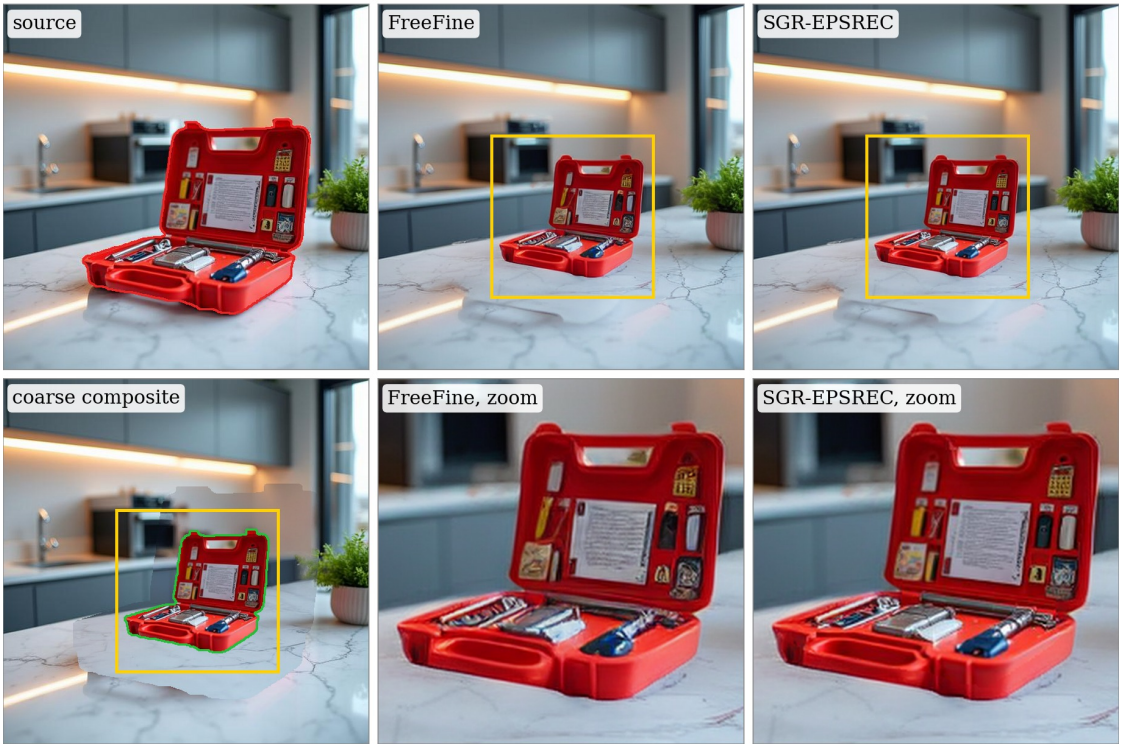


Figure 15.6: Two moderate resizings, chosen and laid out as in Figure 15.4. Change in subject consistency: (e) +0.0081, (f) +0.0028.

Chapter 16

Discussion and Limitations

The experimental chapters each answered one question. This chapter puts the answers together: what the six hypotheses establish as a whole, how much room for improvement FreeFine left, what the metrics can and cannot show, and where the results stop.

16.1 What the six hypotheses establish

Table 16.1 lists the six hypotheses. Read together, they locate the failure found in Chapter 7 and explain the shape of the final method.

Tuning and added information do not explain the failure. The first three hypotheses located the failure outside refinement itself: in its configuration (H1), or in information it is never given: what the object is (H2) and how it was transformed (H3). All three were rejected in the forms that were tested, H2 and H3 through every entry point that a frozen U-Net offers. These experiments cannot exclude every possible setting or every possible way of supplying information, but a likely reason for their outcome is at hand: the coarse composite already shows the object, copied from the source, at its exact target pose, so refinement is not short of information. The evidence points instead to how refinement treats that composite, and the last three hypotheses show that the answer depends on the transformation.

- **Translations and moderate resizings.** The composite's interior is exact or nearly so, and refinement degrades it slightly. Keeping the interior and refining only a thin band along the outline improves consistency (H4).

- **Rotations.** Refinement repairs the composite: on hard rotations it lowers mean distance from 17.86 to 14.01 pixels (Experiment 13), because keypoints are located more reliably in a refined, natural-looking image than in the resampled copy. Keeping the composite undoes that repair, and nothing tested improves on refinement as released.

- **Severe resizings.** The composite's interior is resampled so strongly that it offers little worth keeping (Experiment 17), but its geometry is better than the refined output's (16.44 against 19.47 pixels on hard resizings): refinement moves the rescaled content away from where the transformation put it. Here the reconstruction has to be steered, and guidance with the object label in the form of EPSREC does so (H5).

Table 16.1: The six hypotheses of the thesis and their outcomes.

Hypothesis	Exp.	Outcome	Decisive evidence
H1: a better setting of the released controls	3–7	rejected	no setting raises subject consistency on a hard group by more than 0.0023
H2: the model lacks the object's identity	8	rejected	identity tokens change hard-group subject consistency by between −0.0021 and +0.0005
H3: the model lacks the geometry of the edit	9–11	rejected	no channel raises hard-group subject consistency by more than 0.0014; geometry inside the attention lowers it by up to 0.019
H4: refinement loses what the composite contains	12–17	holds selectively	the output is within 1% of the composite's subject consistency; restoring the interior helps translations and moderate resizings and harms rotations
H5: severe resizings need a better-steered reconstruction	18–21	holds for severe resizings	EPSREC lowers mean distance on hard resizings from 19.47 to 16.61 pixels on the development subset, and by 0.34 pixels on all 759 severe resizings
H6: the composite's structure should be supplied inside the object	22–23	not supported beyond EPSREC	adds nothing separable to EPSREC; lowers mean distance only together with label guidance, an interaction with guidance

Steer the surroundings, not the object. One pattern runs through Chapters 10 to 13. The mechanisms that constrain the object's structure directly while it is denoised (feature maps at the target pose, geometry inside the attention transplant, spectral fusion without guidance) harm it, and the more directly they constrain it, the more they harm. The mechanisms that help act elsewhere: guidance steers the band around the object, and interior restoration keeps the reliable part of the composite after refinement is complete. Spectral fusion shows both sides at once: the same transfer of structure into the object helps when guidance reconstructs the surroundings consistently with it, and harms when it does not (H6). A plausible reading is that the frozen model produces a coherent object only when the object and its surroundings agree, and that forcing the object's interior against the model's own denoising creates disagreements that it cannot resolve.

No treatment suits every edit. Every component that helped one kind of edit harmed another when applied to all of them (Experiments 16 and 21): interior restoration raises subject consistency on rotations while worsening their mean distance by 1.7 to 3.9 pixels, and label guidance helps severe resizings while costing rotations and mild resizings. An average over all edits hides these opposite effects. This is why the final method is a router, and it is a finding in its own right: a training-free change to geometric editing has to be evaluated, and chosen, per kind of transformation.

16.2 How much room there was

The coarse composite, scored as if it were an output (Experiment 13), measures how much of the failure could be recovered at all. It contains nothing but the source object moved to its target pose, so whatever it loses on a metric is lost to the transformation, not to editing.

Subject consistency. On translations the composite scores 0.97; on hard rotations, 0.86. Since the composite is a copy of the source, most of that drop is the response of the metric to the transformation itself: a global descriptor of a rotated or rescaled object differs from that of the original, however faithfully it is copied. The low subject consistency that Chapter 7 measured on hard edits is therefore largely a property of the measurement. FreeFine reaches 99% of the composite's value on hard rotations, which left 0.0085 to 0.0088 to gain on them and 0.0017 to 0.0086 on hard resizings. The threshold of 0.005 set in Chapter 9 lay close to the whole of that room. The only change that reached it was copying the composite's interior into hard rotations (+0.0051, Experiment 17), at a cost of 3.9 pixels in mean distance; on the 800 hard resizings of the whole benchmark the final method raises subject consistency by 0.0010, a value computed from the per-edit differences of §15.3. Mean distance contains a similar share of measurement: the target of each keypoint is computed about a different centre from the one the transformation uses, which charges exactly executed rotations and rescalings a residual that grows with severity (§7.2).

Where the room was. The clearest room was in mean distance on hard resizings, where refinement degrades what the composite gets right, by 3.0 and 1.6 pixels on the two subsets of Experiment 13. It is also the one place where the thesis produced a gain by generating differently rather than by keeping the composite: EPSREC brought hard resizings on the development subset to within 0.17 pixels of the composite, and at full scale it lowers mean distance on severe resizings by 0.34 pixels. The gains of SGR-EPSREC are small in absolute terms, but they should be read against this room: FreeFine is close to what its own input allows on most of the benchmark, and the method improves it where it is not.

16.3 What the metrics can show

Much of the analysis in this thesis went into the metrics, because the seven numbers of GeoBench do not all measure what their names suggest. Three properties shaped the conclusions.

Consistency and realism pull in opposite directions. The composite has the best consistency and the worst realism of any output; refinement converts part of its consistency into realism (Experiment 13). Every method sits somewhere on this trade-off, and the warping error, which is zero for the composite, rewards copying by definition. A method can therefore improve some metrics simply by moving along the trade-off. The components of SGR-EPSREC were chosen by the metric that measures the failure each addresses (§14.1), and the gains that follow from construction were reported but not counted as evidence (§15.3).

Mean distance mixes geometry with appearance. Keypoints are located with diffusion features, which locate them more reliably in a natural-looking image than in a composited one: the composite of a translation is geometrically exact and still scores a worse mean distance than the refined output (Experiment 13). Each method is also scored on its own keypoints, so the value of a group depends on how many keypoints each kind of edit contributes, which is why the whole-benchmark change of SGR-EPSREC is a weighting effect (§15.3). Mean distance was therefore compared only within a group and between methods, and its geometric meaning is clearest where appearance changes little between the methods compared.

Small differences need many edits. On 200 edits the difference in FID between the same two models changes sign between the development and the held-out subset (Experiment 14), and mean distance varies by one to two pixels between near-identical variants (Experiment 3). The development studies could establish directions; the size of an effect was established only on the full benchmark, where the severe-resizing gain turned out smaller than on the edits it was developed on.

16.4 Limitations

Of the method.

- **Rotations are not improved.** No component tested helped them, and SGR-EPSREC leaves them exactly as FreeFine produces them. Hard rotations remain one of the two groups where FreeFine is weakest.

- **Translations pay in mean distance.** Interior restoration raises their consistency at a cost of 5% in mean distance.

- **The geometric gain is small.** On severe resizings mean distance falls by 1.8% at full scale, and the gain in subject consistency is only just significant.

- **EPSREC needs the object's label.** The benchmark provides it; in an editing tool it would have to come from the user or from a captioning model, which was not tested. Without a label the guidance is inert (Experiment 21).

- **Interior restoration uses the benchmark's composite.** The final run took the interior from the composite distributed with GeoBench, whereas a deployed method would use the composite that FreeFine builds for the edit. The two differ in the background, which each fills with its own inpainting. Inside the object they were identical, pixel for pixel, on all 47 edits compared for the figures of this thesis, but they were not compared on every edit of the benchmark.

- **The routing needs the parameters of the edit.** The router reads the type and scale of the transformation. This holds for any edit specified by parameters, as in GeoBench, but not for edits given in another form, such as a dragged point or a text instruction.

Figure 16.1 shows two edits on which the first and third of these limitations can be seen. On the rotation the output is FreeFine's; the transformation turns everything inside the mask, so the falling

water now leaves the spout at an angle, while the ripples it made remain where the stream used to land. On the severe reduction the two outputs are almost identical: both keep the pale silhouette of the original chair that the inpainting stage leaves on the wall and the floor, and both deform the chair's legs.

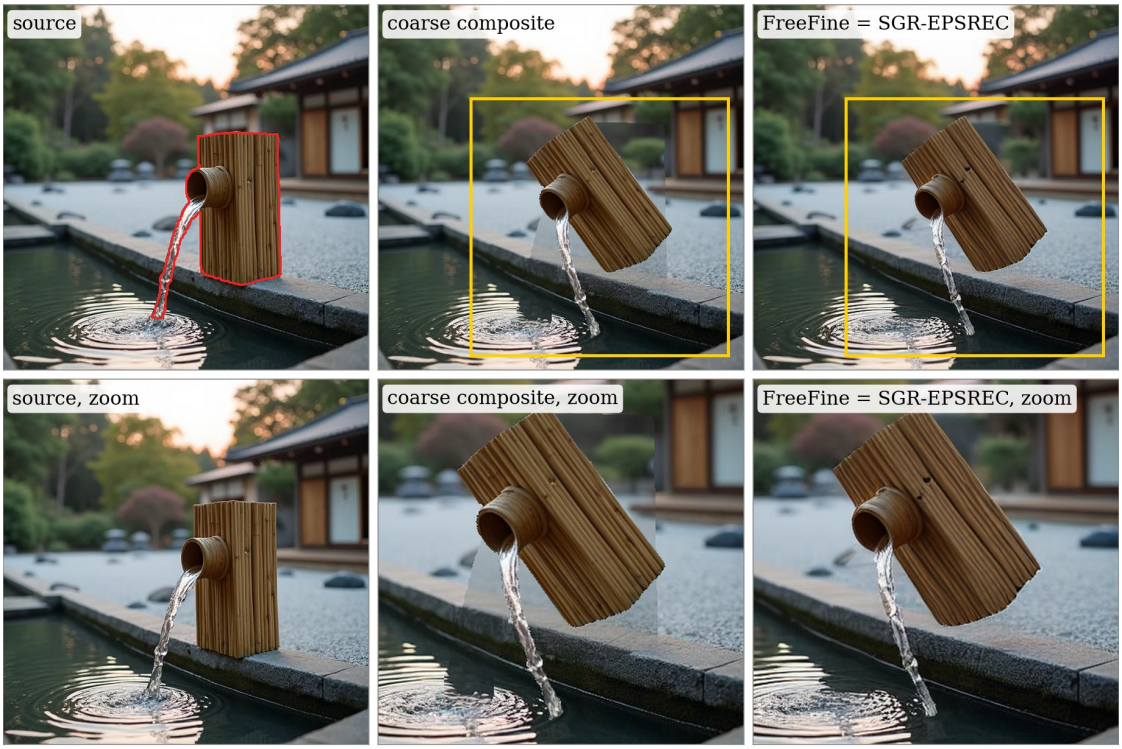
Of the study.

- **One benchmark.** All results concern the two-dimensional, in-plane edits of GeoBench-2D.
- **One seed per edit.** As in the published evaluation of FreeFine, every edit was generated once, with a fixed seed; how much a result varies across seeds was not measured.
- **Development on** 200 **edits.** Components were selected among many variants on the development subset, which favours the variant whose measured gain was helped by chance. The full evaluation shows that the result does not depend on those edits, but also that the gain of the selected guidance is smaller at scale.
- **No perceptual evaluation.** The conclusions rest on the benchmark's metrics; no human study compared the outputs.
- **No paired test for mean distance.** Mean distance was not recorded per edit in the final evaluation, so its changes have no confidence interval.
- **Unexplained deviations of the reproduction.** The reproduced FreeFine scores better than published on the warping error and mean distance (Experiment 1). The cause was not identified; every comparison in the thesis is therefore made against the reproduction, never against the published values.

16.5 Generality

The findings concern one method, FreeFine on Stable Diffusion 1.5, and one benchmark. Two of them are likely to hold more widely, because they depend on the structure of the problem rather than on FreeFine's details. Any editor that builds a coarse composite and then refines it faces the same question of what to keep and what to regenerate, and the answer found here (keep the composite where it is exact, regenerate where it is resampled or rotated, and steer the reconstruction where the transformation is severe) depends on the transformation, not on the network. And the lesson of the metrics, that consistency, precision and realism reward different outputs and must be read per kind of edit, applies to any method evaluated on GeoBench. Whether the specific components transfer (to other backbones, to the three-dimensional edits of GeoBench-3D, or to edits specified by drag or text) was not tested; Chapter 17 returns to these directions.

(a) Hard rotation: 35 degrees (routed to FreeFine unchanged)



(b) Severe resizing: to 0.49 of the original size

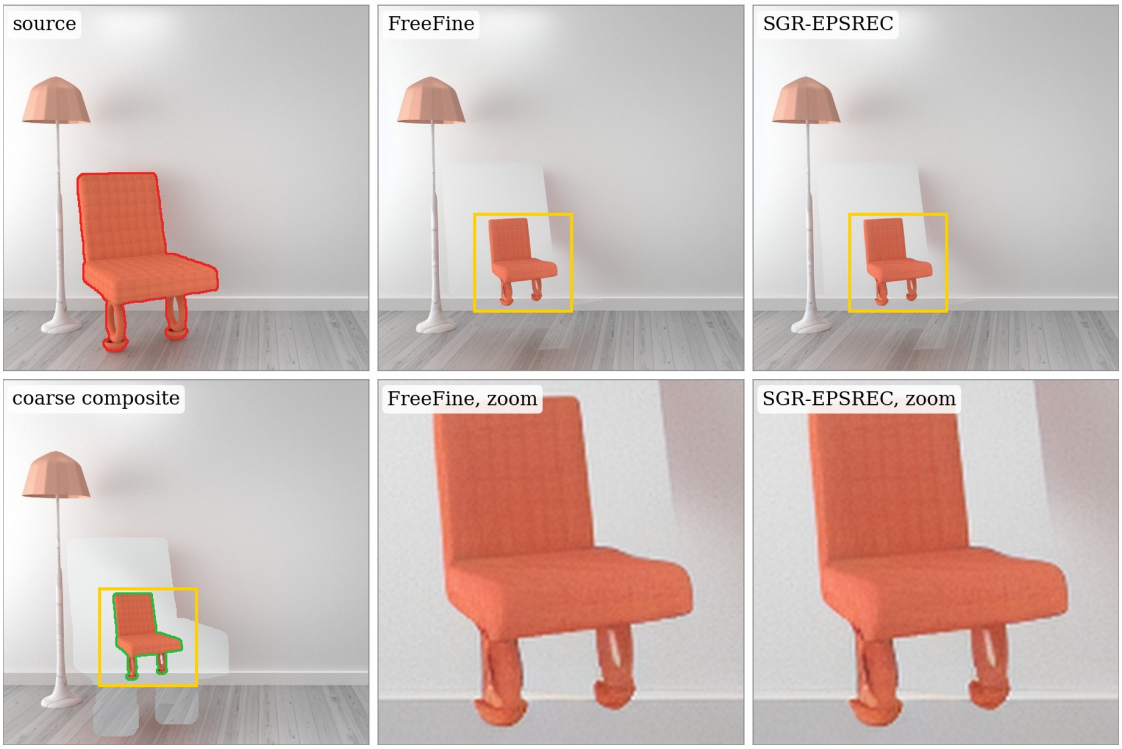


Figure 16.1: Two edits that SGR-EPSREC does not improve. (a) A hard rotation, which the router passes to FreeFine unchanged, laid out as in Figure 7.6: source, coarse composite and output, whole and zoomed. (b) A severe reduction, laid out as in Figure 15.1; its change in subject consistency is −0.0002.

Chapter 17

Conclusions and Future Work

17.1 Conclusions

This thesis set out to understand where FreeFine, a training-free method for geometric image editing, fails, and to improve it without training. It reached six conclusions.

1. **FreeFine can be reproduced faithfully.** Run from its public code on GeoBench-2D, it matches the published realism and consistency to within 2% and scores better on the warping error and mean distance (Chapter 7). Generation is deterministic, so every later comparison could be made image by image against the same reproduced baseline.

2. **The failure is concentrated in hard rotations and hard resizings.** These edits, 26% of the benchmark, carry most of the loss in subject consistency and geometric precision, while background consistency and the warping error are flat across edits (Chapter 7). Part of this failure belongs to the metrics rather than to the method: the coarse composite, a copy of the source at its target pose, already loses most of the subject consistency that hard edits lose, and FreeFine reaches 99% of the composite's value (Chapter 16).

3. **None of the tested reconfigurations or identity- and geometry-conditioning mechanisms explains the failure.** No tested setting of the released controls, and no tested representation of the object's identity or of the edit's geometry, supplied through each entry point of the frozen U-Net, raised subject consistency on a hard group appreciably (Hypotheses 1 to 3, Experiments 3 to 11). These experiments do not exclude every possible configuration or conditioning mechanism; they show that the tested ones do not help. The evidence points instead to how refinement treats the composite, which already shows the object at its target pose.

4. **How refinement should treat the composite depends on the transformation.** Keeping the composite's interior improves translations and moderate resizings; refinement as released is best for rotations, which it repairs; and severe resizings need their reconstruction steered, which guidance with the object label achieves in the form of EPSREC, a projected guidance with recursive momentum developed in this thesis (Hypotheses 4 and 5, Experiments 12 to 21). Supplying the composite's structure inside the object lowers mean distance only together with that guidance and adds nothing separable to EPSREC, so Hypothesis 6 is not supported. Mechanisms that constrain the object directly while it is denoised harm it; those that steer its surroundings, or keep what is reliable after refinement, help.

5. **SGR-EPSREC improves FreeFine on the whole benchmark.** The router sends each edit to the branch that suits it, using only the requested transformation, and changes nothing that the earlier experiments did not establish (Chapter 14). On all 5,677 edits it improves all seven metrics of GeoBench-2D, raises subject consistency for 54% of the edits and lowers it for 18%, lowers mean distance on severe resizings by 0.34 pixels, leaves rotations exactly as FreeFine produces them, and does not lower realism (Chapter 15).

6. **The metrics of the benchmark must be read with care.** Consistency and realism reward opposite outputs; the warping error rewards copying the composite by definition; mean distance mixes geometry with appearance and depends on how many keypoints each edit contributes; and the realism metrics can compare methods only on the same group of edits. Several of the seven improvements of the final method follow partly from these properties, and the thesis separates them from the improvements that reflect better editing (Chapters 15 and 16).

Taken together, the thesis shows that FreeFine is close to what its own input allows on most of GeoBench-2D, and that the remaining room is recovered not by one global change but by treating each transformation according to what its coarse composite gets right: keep it, leave refinement alone, or steer the reconstruction.

17.2 Future work

The thesis worked within deliberate limits: a frozen model, changes that need no training, and the in-plane edits of GeoBench-2D. The directions below lie beyond those limits, and each follows from one of its findings.

Rotations beyond the training-free setting. Every training-free change tested on rotations (more information about the object, its exact geometry, anchoring to the composite, guidance) either left them unchanged or made them worse on one of the two metrics that measure the failure. The evidence points to a limit of the frozen model itself: it reconstructs a rotated object as well as its own features allow, and supplying more information does not change those features. Improving rotations is therefore likely to require changing the model, for example by training a small component dedicated to in-plane rotation while keeping the rest of the pipeline frozen. This leaves the training-free setting, and it is the natural next step for the one kind of edit this thesis could not improve.

A metric for geometric editing. A recurring finding was that the benchmark's metrics do not measure only what their names suggest: subject consistency responds to the transformation itself, the warping error rewards copying, and mean distance mixes geometry with appearance and with the weighting of keypoints. A metric that measures where content ends up independently of how it looks, that does not penalise a correctly executed transformation, and that is validated against human judgement would let methods for geometric editing be compared on what they are meant to do. Designing and validating such a metric is a contribution to the field independent of any one method.

Out-of-plane edits. GeoBench also contains three-dimensional edits that rotate objects out of the image plane. There the composite cannot show the object at its target pose, because parts of it never appeared in the source, so the central question of this thesis changes: what to keep of the composite is no longer enough, and the missing content must come from a prior about the object's three-dimensional shape. How such a prior should be combined with what the composite does get right is open.

Editing the scene instead of the object. Replacing the background around an object raises the question of this thesis from the other side (the object must be kept exactly and the scene regenerated) and adds problems that geometric editing on GeoBench does not pose: the object's lighting and shadows must agree with the new scene, and there is no ground truth for the result. Datasets for image harmonisation, which pair composited images with real photographs, could provide the evaluation that such a task needs.

Newer generative models. All results were obtained on Stable Diffusion 1.5. Newer models, including transformer-based ones trained with flow matching, reconstruct fine structure better and parameterise denoising differently: EPSREC was derived for a model that predicts noise, and its form would have to be derived again for one that predicts velocity. Whether the failure on hard edits, and the room for improvement measured here, survive on such models is open.

Bibliography

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning (ICML)*, 2015.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations (ICLR)*, 2021.

[4] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[5] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021.

[6] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, "Null-text inversion for editing real images using guided diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[12] R. Rombach and P. Esser, "Stable diffusion v1-5 model card." Hugging Face model repository, https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5, 2022.

[13] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[15] M. Cao, X. Wang, Z. Qi, Y. Shan, X. Qie, and Y. Zheng, "MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
arXiv:2304.08465.

[16] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross-attention control," in *International Conference on Learning Representations (ICLR)*, 2023.

[17] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, "Plug-and-play diffusion features for text-driven image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[18] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[19] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[20] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, "DiffEdit: Diffusion-based semantic image editing with mask guidance," in *International Conference on Learning Representations (ICLR)*, 2023.

[21] Y. Alaluf, D. Garibi, O. Patashnik, H. Averbuch-Elor, and D. Cohen-Or, "Cross-image attention for zero-shot appearance transfer," in *ACM SIGGRAPH Conference Proceedings*, 2024.

[22] T. Brooks, A. Holynski, and A. A. Efros, "InstructPix2Pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] X. Pan, A. Tewari, T. Leimkühler, L. Liu, A. Meka, and C. Theobalt, "Drag your GAN: Interactive point-based manipulation on the generative image manifold," in *ACM SIGGRAPH Conference Proceedings*, 2023.

[24] Y. Shi, C. Xue, J. H. Liew, J. Pan, H. Yan, W. Zhang, V. Y. F. Tan, and S. Bai, "DragDiffusion: Harnessing diffusion models for interactive point-based image editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
arXiv:2306.14435.

[25] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[26] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang, "DragonDiffusion: Enabling drag-style manipulation on diffusion models," in *International Conference on Learning Representations (ICLR)*, 2024.
arXiv:2307.02421.

[27] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang, "DiffEditor: Boosting accuracy and flexibility on diffusion-based image editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
arXiv:2402.02583.

[28] J. Lu, X. Li, and K. Han, "RegionDrag: Fast region-based image editing with diffusion models," in *European Conference on Computer Vision (ECCV)*, 2024.

[29] D. Epstein, A. Jabri, B. Poole, A. A. Efros, and A. Holynski, "Diffusion self-guidance for controllable image generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[30] D. Geng and A. Owens, "Motion guidance: Diffusion-based image editing with differentiable motion estimators," in *International Conference on Learning Representations (ICLR)*, 2024.

[31] R. Sajnani, J. Vanbaar, J. Min, K. Katyal, and S. Sridhar, "GeoDiffuser: Geometry-based image editing with diffusion models," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.
arXiv:2404.14403.

[32] K. Pandey, P. Guerrero, M. Gadelha, Y. Hold-Geoffroy, K. Singh, and N. Mitra, "Diffusion handles: Enabling 3d edits for diffusion models by lifting activations to 3d," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
arXiv:2312.02190.

[33] Y. Jia, Y. Yuan, A. Cheng, C. Wang, J. Li, H. Jia, and S. Zhang, "DesignEdit: Multi-layered latent decomposition and fusion for unified and accurate image editing," *arXiv preprint arXiv:2403.14487*, 2024.

[34] H. Zhu, Z. Zhu, K. Zhang, Y. Gong, Y. Liu, and X. Bai, "Training-free geometric image editing on diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.
arXiv:2507.23300.

[35] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[36] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[37] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, Y. Shan, and X. Qie, "T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," in *AAAI Conference on Artificial Intelligence*, 2024.

[38] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models," *arXiv preprint arXiv:2308.06721*, 2023.

[39] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[40] T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen, "Applying guidance in a limited interval improves sample and distribution quality in diffusion models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[41] H. Chung, J. Kim, G. Y. Park, H. Nam, and J. C. Ye, "CFG++: Manifold-constrained classifier free guidance for diffusion models," in *International Conference on Learning Representations (ICLR)*, 2025.

[42] S. Sadat, O. Hilliges, and R. M. Weber, "Eliminating oversaturation and artifacts of high guidance scales in diffusion models," in *International Conference on Learning Representations (ICLR)*, 2025.
arXiv:2410.02416.

[43] S. Sadat, T. Vontobel, F. Salehi, and R. M. Weber, "Guidance in the frequency domain enables high-fidelity sampling at low CFG scales," *arXiv preprint arXiv:2506.19713*, 2025.

[44] C. Si, Z. Huang, Y. Jiang, and Z. Liu, "FreeU: Free lunch in diffusion U-Net," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[46] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *International Conference on Learning Representations (ICLR)*, 2018.

[47] G. Stein, J. C. Cresswell, R. Hosseinzadeh, Y. Sui, B. L. Ross, V. Villecroze, Z. Liu, A. L. Caterini, J. E. T. Taylor, and G. Loaiza-Ganem, "Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
arXiv:2306.04675.

[48] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[49] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[51] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[52] P. S. Heckbert, "Fundamentals of texture mapping and image warping," Master's thesis, University of California, Berkeley, 1989.
Technical Report UCB/CSD-89-516.

[53] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[54] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, 2022.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[56] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.