



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Arthur Gislon Leonida

Extensão do DreamCatalyst para Edição 3D Orientada por Texto: direcionamento adaptativo e localização espacial em campos de radiância neurais

Florianópolis
2026

Arthur Gislon Leonida

Extensão do DreamCatalyst para Edição 3D Orientada por Texto: direcionamento adaptativo e localização espacial em campos de radiância neurais

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Laio Oriel Seman, Dr.

Supervisor: Prof. Laio Oriel Seman, Dr.

Florianópolis

2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Leonida, Arthur Gislou

Extensão do DreamCatalyst para Edição 3D Orientada por
Texto : Direcionamento adaptativo e localização espacial em
campos de radiância neurais / Arthur Gislou Leonida ;
orientador, Laio Oriel Seman, 2026.

90 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2026.

Inclui referências.

1. Engenharia de Controle e Automação. 2. IA Generativa.
3. Modelos de Difusão. 4. NeRF. 5. Visão Computacional. I.
Seman, Laio Oriel. II. Universidade Federal de Santa
Catarina. Graduação em Engenharia de Controle e Automação.
III. Título.

Arthur Gislon Leonida

Extensão do DreamCatalyst para Edição 3D Orientada por Texto: direcionamento adaptativo e localização espacial em campos de radência neurais

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 08 de Julho de 2026.

Coordenação do Curso
Curso de Graduação em Engenharia de Controle e Automação

Banca Examinadora:

Prof. Laio Oriel Seman, Dr.
Orientador
UFSC/CTC/EAS



Documento assinado digitalmente

Laio Oriel Seman

Data: 14/07/2026 14:49:16-0300

CPF: ***.236.939-**

Verifique as assinaturas em <https://v.ufsc.br>

Prof. Laio Oriel Seman, Dr.
Supervisor
UFSC/CTC/EAS

Prof. Eric Aislan Antonelo, Dr.
Avaliador
UFSC/CTC/EAS

Prof. Ubirajara Franco Moreno, Dr.
Presidente da Banca
UFSC/CTC/EAS

Este trabalho é dedicado aos professores que me apoiaram ao longo desta jornada, aos meus queridos pais e à minha avó (in memoriam).

AGRADECIMENTOS

Agradeço aos professores, em especial ao professor Marcelo De Lellis Costa de Oliveira por ter me guiado durante a etapa mais importante da minha graduação, ao professor Eric Aislan Antonelo por ter me proporcionado uma base sólida sobre IA e ao professor Laio Oriel Seman por me mostrar uma forte mentalidade de pesquisa e pelas dicas no decorrer do meu projeto. Agradeço também à Universidade Federal de Santa Catarina pela formação em Engenharia de Controle e Automação e pela infraestrutura acadêmica que tornou possível o desenvolvimento deste trabalho, em especial ao VLAB, um laboratório altamente inovador, fundamental para posicionar a UFSC na vanguarda da pesquisa em IA no Brasil.

“Algumas pessoas querem que algo aconteça, outras desejam que aconteça, outras fazem acontecer.”
(Michael Jordan)

DECLARAÇÃO DE PUBLICIDADE

Florianópolis, 30 de Junho de 2026.

Na condição de representante da instituição de realização do PFC na qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que o autor, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa n° 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.



Documento assinado digitalmente

Laio Oriel Seman

Data: 14/07/2026 14:49:39-0300

CPF: ***.236.939-**

Verifique as assinaturas em <https://v.ufsc.br>

Laio Oriel Seman, Dr.
UFSC/CTC/EAS

RESUMO

Este trabalho investiga o problema de edição 3D orientada por texto em cenas reconstruídas a partir de múltiplas imagens. O ponto de partida é o DreamCatalyst, método que utiliza modelos de difusão 2D, em particular o InstructPix2Pix, para supervisionar a otimização de uma representação 3D por meio do *Delta Denoising Score*. Embora essa abordagem permita alterar semanticamente uma cena 3D usando comandos em linguagem natural, ela herda limitações importantes de métodos 2D aplicados vista a vista, como inconsistência entre diferentes ângulos, alterações indesejadas no fundo e dificuldade de localizar a região correta da edição. O objetivo deste trabalho é estender o DreamCatalyst com mecanismos modulares de *guidance*, localização espacial e preservação. A solução proposta combina TAG adaptativo, STG, máscaras derivadas da diferença entre predições de ruído, máscaras semânticas de *Cross-Attention* com peso programado ao longo da otimização, *source-blended localization* entre os *scores* de fonte e alvo e adaptação por força de edição. A implementação foi realizada sobre Nerfstudio e mantém a compatibilidade com os fluxos de reconstrução por NeRF e 3D Gaussian Splatting. Os experimentos mostram que, na configuração universal proposta (Parte 1, o resultado central do trabalho), há um ganho claro de preservação sem perda de editabilidade: a proposta melhora o SSIM nas oito edições avaliadas e reduz o LPIPS em sete delas (em média, SSIM de 0,825 para 0,854 e LPIPS de 0,272 para 0,247), mantendo uma editabilidade (a eficácia da edição em seguir o comando textual, medida pelo *CLIP Direction*) equivalente à do DreamCatalyst base. Como investigação complementar (Parte 2), um *voxel cache* 3D opcional agrega a magnitude de edição observada por cada vista em uma grade de *voxels* e ataca o modo de falha em que a mesma região física é editada em uma vista e preservada em outra. Sua principal conclusão não é uma melhora de métricas (nas métricas agregadas, em oito edições e $n = 3$, o módulo é neutro), mas uma mudança no tipo de erro: as falhas residuais deixam de depender do ângulo de observação e tornam-se coerentes entre vistas, propriedade desejável em edição 3D, ainda que edições de recoloração ampla exijam um *gate* de concordância entre vistas.

Palavras-chave: edição 3D. modelos de difusão. campos de radiância neurais. DreamCatalyst. localização espacial.

ABSTRACT

This work investigates the problem of text-guided 3D editing in scenes reconstructed from multiple images. The starting point is DreamCatalyst, a method that uses 2D diffusion models, particularly InstructPix2Pix, to supervise the optimization of a 3D representation via the Delta Denoising Score. Although this approach enables a 3D scene to be semantically altered using natural language commands, it inherits significant limitations from view-by-view 2D methods, such as multi-view inconsistency, undesired background alterations, and difficulty in localizing the correct editing region. The objective of this undergraduate thesis is to extend DreamCatalyst with modular mechanisms for guidance, spatial localization, and preservation. The proposed solution combines adaptive TAG, STG, masks derived from the difference between noise predictions, semantic Cross-Attention masks with scheduled weighting throughout the optimization, source-blended localization between source and target scores, and editing strength adaptation. The implementation was built on top of Nerfstudio and maintains compatibility with NeRF and 3D Gaussian Splatting reconstruction pipelines. Experiments show that the proposed universal configuration (Part 1, the main result) yields a clear preservation gain at unchanged editability: it improves SSIM on all eight evaluated edits and reduces LPIPS on seven of them (on average, SSIM from 0.825 to 0.854 and LPIPS from 0.272 to 0.247), while keeping editability (the effectiveness of the edit in following the text command, measured by CLIP Direction) comparable to the DreamCatalyst baseline. As a complementary investigation (Part 2), an optional 3D *voxel cache* aggregates the per-view edit magnitude into a voxel grid and targets the failure mode in which the same physical region is edited in one view yet preserved in another. Its main conclusion is not a metric improvement (on the aggregate metrics, over eight edits and $n = 3$, the module is neutral), but a change in the type of error: residual failures stop depending on the viewing angle and become coherent across views, a desirable property for 3D editing, even though broad recoloring edits still require a cross-view agreement gate.

Keywords: 3D editing. diffusion models. neural radiance fields. DreamCatalyst. spatial localization.

LISTA DE FIGURAS

Figura 1 – Estrutura do campo neural do NeRF: as posições amostradas passam pela codificação posicional e por um MLP que prediz densidade; a direção de observação é injetada ao final para produzir a cor dependente da vista.	25
Figura 2 – Esquema simplificado da renderização volumétrica em NeRF. Para cada pixel, um raio é lançado a partir da câmera e amostras 3D são avaliadas ao longo desse raio. A rede neural prediz cor e densidade em cada amostra, e a cor final do pixel é obtida por uma composição ponderada pelas densidades e transmitâncias acumuladas.	26
Figura 3 – Estratégia de amostragem hierárquica do NeRF na implementação do Nerfstudio: um amostrador uniforme cobre a faixa de profundidades e um amostrador por densidade de probabilidade (<i>PDF sampler</i>) concentra as amostras finais nas regiões próximas às superfícies. .	27
Figura 4 – Visão geral do <i>pipeline</i> NeRF como implementado no Nerfstudio: raios são gerados a partir das câmeras, amostrados em dois estágios, avaliados pelo campo neural e compostos pelos renderizadores.	27
Figura 5 – Esquema do processo direto e reverso em modelos de difusão. No processo direto, ruído é adicionado progressivamente a uma amostra limpa x_0 até aproximá-la de ruído puro x_T . No processo reverso, uma rede de remoção de ruído estima como transformar uma amostra ruidosa em uma amostra limpa.	29
Figura 6 – Esquema da arquitetura de remoção de ruído usada pelo InstructPix2Pix. O latente ruidoso da vista renderizada é concatenado ao latente da imagem fonte, enquanto a instrução textual condiciona os blocos de <i>Cross-Attention</i> do UNet.	31
Figura 7 – <i>Cross-Attention Up block</i> . No caminho do decodificador, a <i>Cross-Attention</i> ajuda a transformar a instrução textual em alterações espacialmente localizadas enquanto a resolução latente é recuperada. .	33
Figura 8 – Laço de otimização do DreamCatalyst base: interação entre o NeRF (domínio 3D, parâmetros θ) e o IP2P (domínio 2D, supervisão por difusão). O NeRF renderiza uma vista, que é codificada e ruidificada em z_t ; o IP2P, condicionado pela vista fonte e pelos <i>prompts</i> t_{src}/t_{tgt} , gera as predições ϵ_{tgt} e ϵ_{src} ; a diferença entre elas (Delta Denoising Score (DDS)) retropropaga pelo renderizador diferenciável e atualiza θ , fechando o laço. A supervisão é 2D e por vista, mas é integrada em uma única representação 3D.	36

Figura 9 – Visão pedagógica da extensão modular do DreamCatalyst. O fluxo separa geração do sinal de edição, localização espacial, um módulo exploratório de memória multivista por <i>voxel cache</i> (opcional, avaliado na Parte 2) e preservação antes da atualização DDS da cena 3D.	39
Figura 10 – Visão geral dos módulos de <i>guidance</i> , localização e preservação adicionados ao DreamCatalyst.	41
Figura 11 – Interpretação geométrica de TAG. A componente radial controla a direção alinhada ao latente ruidoso, enquanto a componente tangencial altera a direção efetiva do <i>score</i> sem simplesmente aumentar toda a norma da predição.	43
Figura 12 – Mecanismo original do STG (<i>Spatiotemporal Skip Guidance</i>), de Hyung et al. (Hyung et al., 2025), proposto para modelos de difusão baseados em <i>transformers</i> (originalmente, difusão de vídeo): uma predição completa é comparada com uma predição fraca, obtida ao pular camadas internas do modelo, e a diferença entre os dois ramos reforça direções estruturais do <i>score</i> . Neste trabalho, o mesmo princípio de <i>skip-guidance</i> é adaptado ao UNet do InstructPix2Pix, perturbando suas camadas de atenção.	44
Figura 13 – Padrões de escalonamento (<i>schedule shapes</i>) usados para controlar a escala relativa de STG. O modo <i>bump</i> concentra a amplificação na fase intermediária e libera a fase final para refinamento sem congelar estados parciais dependentes da vista.	45
Figura 14 – Exemplo qualitativo de vista renderizada e mapas de localização usados pelo método.	45
Figura 15 – Mapa de variância entre vistas do <i>voxel cache</i> (esquerda), normalizado por $\sigma_{\max}^2$, ao lado da vista editada correspondente (direita) na cena <i>clown</i> . A cabeça aparece escura (baixa variância entre vistas: edição desejada e coerente), enquanto os braços ficam <i>um pouco mais claros</i> : variância mais alta que a da cabeça, sinalizando a região mais propensa à sobre-edição, ainda que nesta execução ela não tenha ocorrido. O contraste é modesto, mas suficiente para posicionar $\sigma_{\max}^2$ no vão entre as duas regiões; é esse o instrumento de ajuste por cena do <i>gate</i> de concordância.	50

Figura 16 – Motivação visual para o <i>voxel cache</i> 3D: sem agregação 3D, a mesma região física (o braço) pode ser editada em uma vista e preservada em outra; a agregação em <i>voxels</i> compartilha a decisão entre as câmeras. A Parte 2 do Capítulo 6 mostra que o <i>cache</i> de fato restaura essa coerência entre vistas e caracteriza o regime de utilidade resultante.	54
Figura 17 – Curvas de treinamento (WandB) da configuração universal proposta, sem <i>voxel cache</i> : o escalonamento <i>bump</i> de STG, o peso escalonado da máscara de <i>Cross-Attention</i> , o envelope de força de edição ($1 - s$) e o η adaptativo de TAG. A chave WandB de cada curva está indicada no painel correspondente.	62
Figura 18 – Resultados qualitativos da Parte 1 (<i>voxel cache</i> desligado): renderização fonte, DreamCatalyst base e a configuração proposta. A proposta mantém editabilidade semelhante ao método base, com melhor preservação de identidade, textura e regiões fora da edição (complementa a Tabela 3).	63
Figura 19 – Comparação qualitativa entre imagem fonte, DreamCatalyst base e método proposto, em mais de uma vista para evidenciar coerência entre vistas e não apenas a qualidade de uma imagem isolada. . . .	65
Figura 20 – Exemplo qualitativo de preservação de fundo. Na cena original, há um interruptor visível na parede. O DreamCatalyst base remove ou degrada esse detalhe durante a edição, enquanto a configuração universal proposta preserva melhor a região fora do objeto editado.	66
Figura 21 – Comparação qualitativa de preservação global do fundo na cena <i>bear</i> (estátua de urso → panda). Da esquerda para a direita: reconstrução fonte, DreamCatalyst base e configuração universal proposta. A proposta mantém textura, iluminação e cores do fundo próximas às da cena fonte, enquanto o método base suaviza a vegetação e desloca a tonalidade global.	66
Figura 22 – Consistência multivista renderizada na cena <i>elf</i> sob duas estratégias de leitura do <i>voxel cache</i> . Na linha superior (vizinho mais próximo), a discretização em blocos do <i>cache</i> introduz falso desacordo entre as vistas; na linha inferior (trilinear ponderada por observação) a mesma região fica coerente entre os ângulos. A leitura trilinear eleva a <i>MV Pairwise Cos Sim</i> de $\approx 0,924$ para 0,933.	70

Figura 23 – Comparação qualitativa da Parte 2 por cena. Da esquerda para a direita: a reconstrução fonte, a configuração universal (Parte 1, sem <i>voxel cache</i>) e a configuração com <i>voxel cache</i> (Parte 2). O ganho do <i>cache</i> é de coerência <i>entre vistas</i> (Seção 6.3.4), não de aparência em uma vista isolada.	73
Figura 24 – Máscaras na cena <i>clown</i> : a máscara auto-derivada (M_{self}) é nítida sobre o sujeito, enquanto a máscara de <i>Cross-Attention</i> (M_{attn}) é mais difusa. A máscara fundida resulta da modulação multiplicativa, que nunca excede a máscara auto-derivada.	76
Figura 25 – Robustez a <i>guidance</i> elevado na cena <i>stormtrooper</i> , com <i>guidance scale</i> 12,5 nas três imagens. (a) No DreamCatalyst base, o <i>guidance</i> elevado duplica o sujeito e cria um segundo <i>stormtrooper</i> (clone fantasma). (b,c) Tanto a configuração 2D quanto a com <i>voxel cache</i> confinam a edição e não criam clone, ainda sob o mesmo <i>guidance</i> 12,5.	78

LISTA DE TABELAS

Tabela 1	– Cenas e instruções de edição avaliadas, organizadas pelos eixos de dificuldade definidos no Capítulo 5. A coluna <i>Sementes</i> indica o número de execuções por configuração usado em cada cena. . . .	60
Tabela 2	– Configuração universal proposta (módulos contribuídos por este trabalho; <i>voxel cache</i> desligado). Parâmetros herdados do DreamCatalyst base não são listados.	61
Tabela 3	– Resultados quantitativos da Parte 1 (configuração universal) na forma DC → Proposto. Cada célula mostra a <i>média</i> sobre $n = 3$ execuções por cena, com o mesmo <i>checkpoint</i> de reconstrução por cena e uma única configuração fixa (sem ajuste por cena). Destaque em verde = vencedor quando a diferença excede a dispersão combinada das execuções; células sem destaque são empates dentro do piso de ruído. O desvio padrão por cena é pequeno ($\leq 0,013$, exceto <i>fangzhou</i>) e fundamenta o critério de significância adotado; os valores completos de média $\pm$ desvio constam do Apêndice C. . . .	62
Tabela 4	– Parte 2 na forma configuração 2D (Parte 1, sem <i>voxel cache</i>) → configuração com <i>voxel cache</i> (Parte 2, STG 3,0), por cena. Cada célula mostra a <i>média</i> sobre $n = 3$ execuções, mesmo <i>checkpoint</i> por cena. Trata-se de uma comparação <i>entre configurações</i> (a coluna <i>voxel</i> inclui o rebalanceamento de STG), e não de um isolamento do <i>cache</i> . A coluna EMV_{3D} é a <i>EditMaskVariance_3D</i> , métrica de consistência multivista calculada a partir de deltas de RGB renderizado retroprojetados (e não da máscara de treinamento), lida com cautela (menor é melhor apenas nominalmente, pois é confundida pela força de edição; ver a discussão a seguir). Como na Tabela 3, o melhor valor de cada célula aparece em verde; como as duas configurações estão muito próximas, só recebem destaque as diferenças que excedem o desvio-padrão combinado das sementes ($n = 3$). Apenas cinco células se separam do ruído: três a favor do <i>voxel cache</i> (<i>stormtrooper</i> em <i>CLIP Direction</i> ; <i>Einstein</i> e <i>fangzhou</i> em LPIPS) e duas a favor da configuração 2D (<i>bear</i> em <i>CLIP Direction</i> ; <i>Einstein</i> em <i>MV Cos</i>), o que reforça o caráter neutro do módulo no agregado. Os valores completos de média $\pm$ desvio constam do Apêndice D.	69
Tabela 5	– Prompts de fonte (t_{src}) e de alvo (t_{tgt}) usados em cada edição. . . .	87
Tabela 6	– DreamCatalyst base (DC): média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.	88

Tabela 7 – Configuração universal proposta: média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.	88
Tabela 8 – Configuração com <i>voxel cache</i> (Parte 2): média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.	89

LISTA DE ABREVIATURAS E SIGLAS

CFG	Classifier-Free Guidance
CLIP	Contrastive Language-Image Pretraining
COLMAP	Pipeline de Structure-from-Motion e Multi-View Stereo
DDS	Delta Denoising Score
IP2P	InstructPix2Pix
LPIPS	Learned Perceptual Image Patch Similarity
MLP	Multilayer Perceptron
NeRF	Neural Radiance Field
SDS	Score Distillation Sampling
SfM	Structure-from-Motion
SSIM	Structural Similarity Index Measure
STG	Spatiotemporal Skip Guidance
TAG	Tangential Amplified Guidance

SUMÁRIO

1	INTRODUÇÃO	20
1.1	MOTIVAÇÃO	20
1.2	OBJETIVOS	21
1.3	CONTRIBUIÇÕES	22
1.4	ORGANIZAÇÃO DO DOCUMENTO	23
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	REPRESENTAÇÕES 3D DIFERENCIÁVEIS	24
2.1.1	Campos de radiância neurais	24
2.1.2	3D Gaussian Splatting	27
2.1.3	Estrutura a partir de movimento	28
2.2	MODELOS DE DIFUSÃO E EDIÇÃO 2D	28
2.2.1	<i>Classifier-Free Guidance</i>	29
2.2.2	InstructPix2Pix	29
2.2.3	Blocos internos do UNet	30
2.3	<i>SCORE DISTILLATION</i> E EDIÇÃO 3D	32
2.4	LOCALIZAÇÃO ESPACIAL E <i>CROSS-ATTENTION</i>	34
2.5	CONSISTÊNCIA ENTRE VISTAS	34
3	DESCRIÇÃO DO PROBLEMA E REQUISITOS TÉCNICOS	35
3.1	FLUXO DE REFERÊNCIA	35
3.2	FORMULAÇÃO DO PROBLEMA	35
3.3	LIMITAÇÕES OBSERVADAS E RISCOS DO FLUXO DE REFERÊNCIA	37
3.4	REQUISITOS TÉCNICOS	37
3.5	ESCOPO	38
3.6	CRITÉRIOS DE AVALIAÇÃO	38
4	SOLUÇÃO PROPOSTA E METODOLOGIA UTILIZADA	39
4.1	VISÃO GERAL	39
4.2	PREDIÇÕES LIMPAS PARA LOCALIZAÇÃO	42
4.3	DIRECIONAMENTO ADAPTATIVO	42
4.3.1	TAG adaptativo e assimétrico	42
4.3.2	STG com escalonamento temporal	43
4.4	MÁSCARA AUTO-DERIVADA	44
4.5	MÁSCARA SEMÂNTICA POR <i>CROSS-ATTENTION</i>	44
4.6	LOCALIZAÇÃO POR MISTURA FONTE-ALVO	46
4.7	ÂNCORAS DE PRESERVAÇÃO	46
4.8	ADAPTAÇÃO POR FORÇA DE EDIÇÃO	47
4.9	MÁSCARA 3D POR <i>VOXEL CACHE</i>	47
4.9.1	Diversidade angular como fator adicional de confiança	50

4.9.2	Leitura do <i>cache</i> e preparação da máscara 3D	52
4.9.3	Fusão apenas positiva	53
5	METODOLOGIA DE AVALIAÇÃO	55
5.1	PROTOCOLO EXPERIMENTAL	55
5.2	CONFIGURAÇÕES EXPERIMENTAIS	55
5.3	CENAS E <i>PROMPTS</i>	56
5.4	MÉTRICAS	56
5.5	CRITÉRIOS DE INTERPRETAÇÃO	57
5.6	LIMITAÇÕES DA AVALIAÇÃO	57
6	RESULTADOS E DISCUSSÃO	59
6.1	CONFIGURAÇÃO EXPERIMENTAL	59
6.1.1	Configuração universal proposta	60
6.2	PARTE 1 — CONFIGURAÇÃO UNIVERSAL VS DREAMCATALYST .	61
6.2.1	Preservação de regiões fora da edição	65
6.3	PARTE 2 — <i>VOXEL CACHE</i> 3D	67
6.3.1	Trajetória de desenvolvimento: da formulação ingênua ao pacote final	67
6.3.2	Resultados quantitativos por cena	68
6.3.3	Cena de detalhe fino (<i>elf</i>): consistência renderizada	68
6.3.4	Cena de recoloração ampla (<i>clown</i>): o regime de amplificação .	70
6.3.5	Resultados qualitativos da Parte 2	72
6.3.6	Lições sobre os mecanismos da Parte 2	74
6.4	ANÁLISE DOS MECANISMOS DO MÉTODO	74
6.4.1	Trajetórias monotônicas vs não monotônicas	74
6.4.2	A máscara de <i>Cross-Attention</i> é subtrativa por construção . . .	75
6.4.3	A força de edição como envelope de decaimento implícito	76
6.4.4	Balanco entre força de edição e preservação	76
6.4.5	Robustez a <i>guidance</i> elevado: supressão do clone fantasma . .	77
6.5	ESCOPO E LIMITAÇÕES	78
7	CONCLUSÃO	80
7.1	SÍNTESE DO TRABALHO	80
7.2	SÍNTESE DOS RESULTADOS	80
7.3	ESCOPO E LIMITAÇÕES OBSERVADAS	81
7.4	TRABALHOS FUTUROS	82
7.5	CONSIDERAÇÕES FINAIS	82
	REFERÊNCIAS	83
	APÊNDICE A – DETALHES DE REPRODUTIBILIDADE DOS EXPERIMENTOS	86
	APÊNDICE B – PROMPTS DE EDIÇÃO: FONTE E ALVO	87

APÊNDICE C – RESULTADOS DETALHADOS DA PARTE 1 (MÉ- DIA E DESVIO PADRÃO)	88
APÊNDICE D – RESULTADOS DETALHADOS DA PARTE 2 (VO- <i>XEL CACHE</i>)	89

1 INTRODUÇÃO

A edição de cenas 3D orientada por texto busca permitir que um usuário modifique uma cena reconstruída a partir de imagens usando instruções em linguagem natural, por exemplo transformar uma pessoa em um personagem específico, alterar a aparência de um objeto ou preservar o fundo enquanto uma região semanticamente relevante é modificada. Esse problema combina duas linhas recentes de pesquisa: representações neurais de cena, como campos de radiância neurais, e modelos generativos de difusão capazes de interpretar comandos textuais.

O presente trabalho parte do método DreamCatalyst (Shim et al., 2025), que utiliza um modelo de edição 2D, InstructPix2Pix (Brooks; Holynski; Efros, 2023), para supervisionar a otimização de uma representação 3D por meio de uma perda baseada em DDS. A ideia central é renderizar vistas da cena 3D, aplicar um modelo de difusão condicionado pelo comando textual e propagar o gradiente da edição para os parâmetros da representação 3D. A abordagem é poderosa porque permite editar uma cena sem treinar um novo modelo generativo 3D; entretanto, ela também herda problemas de métodos 2D aplicados vista a vista.

De forma intuitiva, a tarefa pode ser vista como editar uma fotografia que não é apenas uma fotografia: cada imagem renderizada é uma janela para uma mesma cena 3D. Se uma instrução textual altera um capacete em uma vista, a mesma alteração deve continuar plausível quando a câmera se move, quando o objeto aparece de lado ou quando uma região antes oculta se torna visível. Por isso, a edição 3D não exige apenas uma boa imagem final em uma câmera; ela exige que o sinal de edição seja integrado de forma coerente pela representação 3D.

1.1 MOTIVAÇÃO

Em uma edição 3D ideal, a mudança desejada deveria ser semanticamente correta, localizada na região relevante e consistente entre múltiplos ângulos de câmera. Na prática, a supervisão por difusão 2D tende a produzir três tipos de problemas. Primeiro, a direção de edição pode ser global, afetando regiões que deveriam permanecer inalteradas. Segundo, a mesma região física da cena pode receber sinais diferentes quando observada por vistas distintas. Terceiro, um aumento agressivo do *guidance* textual pode melhorar a aderência ao *prompt*, mas também gerar saturação, perda de identidade ou alterações no fundo.

Essas limitações agravam-se em cenas 3D porque erros pequenos em imagens individuais podem se acumular durante a otimização da representação. Uma máscara imprecisa em uma única vista pode não parecer problemática isoladamente, mas a otimização 3D integra gradientes de muitas vistas ao longo de milhares de iterações. Portanto, a questão principal deste trabalho é como tornar o processo de edição mais

localizado, preservador e coerente entre vistas, sem abandonar a estrutura geral do DreamCatalyst.

A resposta principal deste trabalho é uma configuração universal de *guidance*, localização e preservação (Parte 1), aqui entendida como uma única configuração fixa, aplicada a todas as cenas avaliadas sem ajuste por cena, e tratada como o resultado central da avaliação. A memória 3D de evidências de edição (*voxel cache*, Parte 2) é uma investigação complementar, voltada especificamente à coerência entre vistas, e não à melhora das métricas agregadas.

Neste texto, a inconsistência entre vistas é tratada como uma vulnerabilidade estrutural de métodos que supervisionam uma cena 3D com sinais 2D calculados por vista. O trabalho não assume que todo resultado do DreamCatalyst original apresente esse problema. Nos experimentos deste trabalho, ele se torna mais evidente quando o *guidance* é fortalecido por módulos como Tangential Amplified Guidance (TAG) e Spatiotemporal Skip Guidance (STG), pois pequenas diferenças de decisão entre câmeras também passam a ser amplificadas. Um exemplo concreto, observado tanto no DreamCatalyst base quanto na configuração universal proposta (sem o *voxel cache*), ocorre na edição que transforma uma pessoa em palhaço: um braço pode aparecer pintado de vermelho quando observado de um ângulo e com tom de pele quando observado de outro, ou seja, a mesma região física da cena recebe decisões de edição opostas conforme a câmera. A memória 3D de evidências de edição (*voxel cache*) investigada neste trabalho foi concebida para atacar exatamente esse modo de falha. A avaliação indica que ela cumpre esse papel no regime para o qual foi concebida: com o *cache*, a decisão de edição passa a ser compartilhada pelas câmeras, e o modo de falha residual muda de natureza. Em parte das execuções, ambos os braços são pintados *consistentemente em todas as vistas*; em outras, nenhum. Quando ocorre, o artefato é coerente em 3D. Isso é preferível a um artefato que muda com o ângulo de observação, mas ainda não é o ideal: nas edições de recoloração ampla, a taxa do defeito indesejado permanece no patamar do processo 2D. O regime de utilidade, portanto, depende da cena. Nas métricas agregadas o módulo é neutro. Seu mecanismo de consistência renderizada se evidencia em isolamento controlado, em edições de detalhe fino. E edições de recoloração ampla exigem *gates* de concordância entre vistas, para que a coerência promovida pelo *cache* não amplifique também sinais coerentes e indesejados.

1.2 OBJETIVOS

O objetivo geral deste trabalho é estender o DreamCatalyst com mecanismos modulares de *guidance*, localização e preservação para edição 3D orientada por texto. Os objetivos específicos são:

- estudar o fluxo completo de reconstrução e edição 3D baseado em COLMAP, Nerfstudio, NeRF/3D Gaussian Splatting e DreamCatalyst;
- implementar variantes de *guidance* que aumentem a força da edição sem comprometer a estrutura da cena;
- construir máscaras de relevância espacial a partir da diferença entre previsões de ruído e de mapas de *Cross-Attention* do modelo de difusão;
- aplicar a máscara diretamente no espaço dos *scores* de difusão por meio de mistura localizada entre fonte e alvo;
- adicionar termos de preservação para reduzir alterações fora da região editada;
- investigar criticamente um *voxel cache* 3D como tentativa de promover consistência entre vistas, avaliando sua eficácia;
- organizar uma metodologia de avaliação quantitativa e qualitativa para comparar a extensão proposta com o DreamCatalyst base.

1.3 CONTRIBUIÇÕES

As principais contribuições técnicas deste trabalho organizam-se em torno de três objetivos. Para **aumentar a força da edição**: TAG adaptativo e assimétrico e STG com escalonamento temporal. Para **localizar** onde a edição ocorre: a máscara auto-derivada, a máscara semântica por *Cross-Attention*, a *source-blended localization* e o *voxel cache* 3D opcional (Parte 2). Para **preservar** o restante da cena: as âncoras de preservação e a adaptação por força de edição. Em detalhe, e incluindo uma caracterização honesta, por regime de cena, do módulo de consistência 3D, as contribuições são:

- um conjunto modular de extensões de *guidance* sobre o DDS, com TAG adaptativo e assimétrico e STG com escalonamento temporal;
- uma máscara auto-derivada baseada em $\|\epsilon_{\text{tgt}} - \epsilon_{\text{src}}\|$, calculada a partir de previsões de ruído antes de aplicar os mecanismos de *guidance*;
- uma máscara semântica baseada em *Cross-Attention*, usada como *prior* textual para restringir a máscara auto-derivada;
- uma formulação de *source-blended localization*, na qual a previsão de ruído alvo é interpolada em direção à previsão fonte fora da máscara;
- uma âncora de preservação fora da máscara e uma âncora simples para controlar deslocamentos de média latente;

- um mecanismo de adaptação por força de edição, calculado diretamente a partir das predições de ruído;
- como investigação complementar (Parte 2), um *voxel cache* 3D de evidências de edição multivista, voltado à coerência *entre vistas* e não à melhora das métricas agregadas. Com o *cache*, a decisão de edição passa a ser compartilhada pelas câmeras e as falhas residuais tornam-se coerentes em 3D, deixando de depender do ângulo de observação. O módulo inclui sua infraestrutura completa (agregação por profundidade e raios de câmera, *gates* de confiança, variância com ponderação exponencial e leitura trilinear ponderada por observação) e uma métrica dedicada (*EditMaskVariance_3D*). Sua caracterização é honesta e por regime de cena: neutro nas métricas agregadas, com editabilidade no piso de ruído na maioria das cenas, necessidade de *gates* de concordância em recoloração ampla, e documentação dos modos de falha encontrados no caminho (descasamento de escala, deriva temporal da variância e realimentação do *gate* de concordância).

1.4 ORGANIZAÇÃO DO DOCUMENTO

O Capítulo 2 apresenta a fundamentação teórica necessária para compreender representações 3D, modelos de difusão, Classifier-Free Guidance (CFG), Score Distillation Sampling (SDS), DDS, InstructPix2Pix e mecanismos de atenção. O Capítulo 3 formaliza o problema tratado e os requisitos técnicos. O Capítulo 4 descreve a solução proposta e seus componentes metodológicos. O Capítulo 5 detalha a metodologia de avaliação. O Capítulo 6 apresenta e discute os resultados quantitativos e qualitativos. Por fim, o Capítulo 7 resume as contribuições, o escopo, as limitações observadas e os próximos passos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos utilizados ao longo do trabalho. A Seção 2.1 discute representações 3D diferenciáveis e a Seção 2.2 resume modelos de difusão e InstructPix2Pix. Em seguida, a Seção 2.3 descreve *score distillation*, SDS e DDS. A Seção 2.4 introduz máscaras de localização e *Cross-Attention*, e a Seção 2.5 discute a consistência entre vistas.

O fio condutor do capítulo é o seguinte: primeiro, é necessário representar uma cena de modo diferenciável; em seguida, obter de um modelo de difusão um sinal que indique como uma vista renderizada deveria mudar; por fim, localizar esse sinal e compartilhá-lo entre vistas. Os conceitos a seguir, portanto, não são componentes isolados, mas peças de uma mesma cadeia de otimização.

2.1 REPRESENTAÇÕES 3D DIFERENCIÁVEIS

Uma representação 3D diferenciável permite renderizar uma imagem a partir de parâmetros de cena e propagar gradientes da imagem renderizada de volta para esses parâmetros. Essa propriedade é essencial em métodos de edição por difusão, pois o modelo 2D fornece um gradiente no espaço da imagem ou do latente, enquanto a variável otimizada é a cena 3D.

2.1.1 Campos de radiância neurais

Neural Radiance Fields (NeRF) representam uma cena contínua por meio de uma função neural (Mildenhall et al., 2020)

$$F_{\theta} : (\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma), \quad (1)$$

na qual $\mathbf{x} \in \mathbb{R}^3$ é uma posição no espaço, $\mathbf{d}$ é a direção de observação, $\mathbf{c}$ é a cor emitida e σ é a densidade volumétrica. A intuição é que a rede responde, para cada ponto do espaço, a duas perguntas: “quanta matéria existe aqui?” (a densidade σ , que funciona como uma opacidade local) e “que cor esse ponto emite quando observado desta direção?” (a cor $\mathbf{c}$, dependente da vista). A dependência da direção de observação é o que permite ao modelo reproduzir efeitos não lambertianos, como reflexos especulares, que variam conforme a câmera se move. O NeRF assume cenas estáticas, iluminação constante e poses de câmera conhecidas; no fluxo deste trabalho, essas poses são estimadas previamente por Structure-from-Motion (SfM), descrito ao final desta seção.

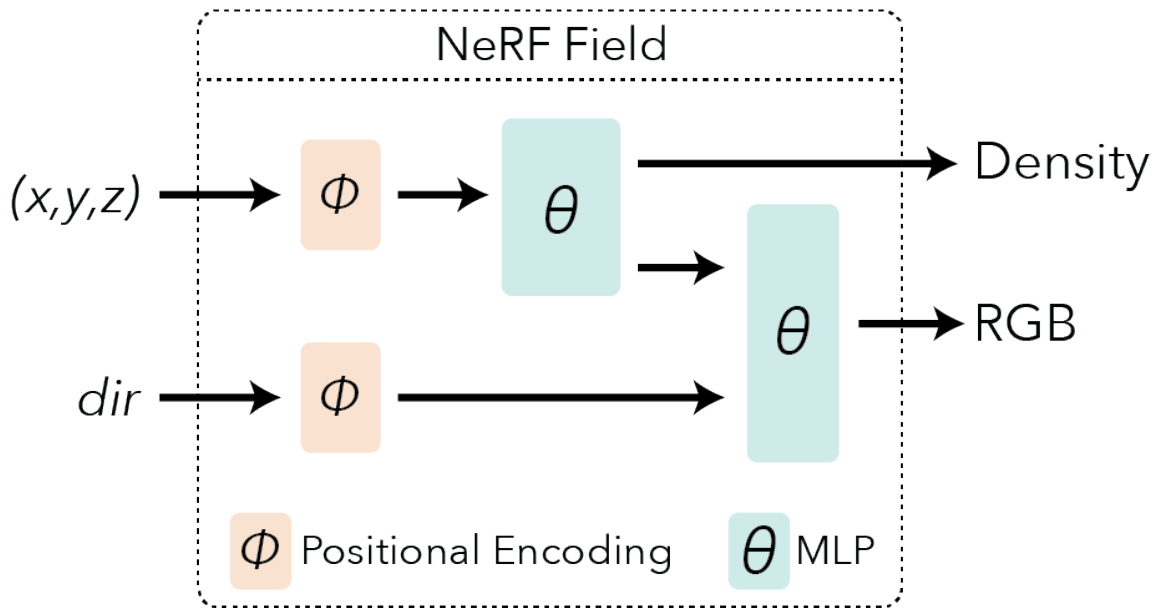
Codificação posicional. Redes Multilayer Perceptron (MLP) treinadas diretamente sobre coordenadas $(\mathbf{x}, \mathbf{d})$ tendem a aprender apenas variações suaves (baixas frequências), borrando detalhes finos de textura e geometria. Por isso, antes de entrar na rede, cada coordenada escalar p é expandida em uma série de senos e cossenos

de frequências crescentes (Mildenhall et al., 2020):

$$\gamma(p) = \left(\sin(2^0 \pi p), \cos(2^0 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p) \right), \quad (2)$$

na qual L é o número de frequências ($L=10$ para posição e $L=4$ para direção na implementação de referência (Nerfstudio Team, 2023)). Essa expansão fornece à rede “régua” de várias escalas: as frequências baixas capturam a forma global da cena e as altas permitem representar bordas e texturas finas. A Figura 1 resume a estrutura do campo neural, da codificação posicional à cor dependente da direção de observação.

Figura 1 – Estrutura do campo neural do NeRF: as posições amostradas passam pela codificação posicional e por um MLP que prediz densidade; a direção de observação é injetada ao final para produzir a cor dependente da vista.



Fonte: Documentação oficial do Nerfstudio (Nerfstudio Team, 2023).

Renderização volumétrica. Para uma câmera com origem o e direção de raio d , os pontos ao longo do raio são dados por

$$r(t) = o + td. \quad (3)$$

A cor renderizada é obtida por integração volumétrica, formulação herdada da renderização de densidades volumétricas (Kajiya; Von Herzen, 1984):

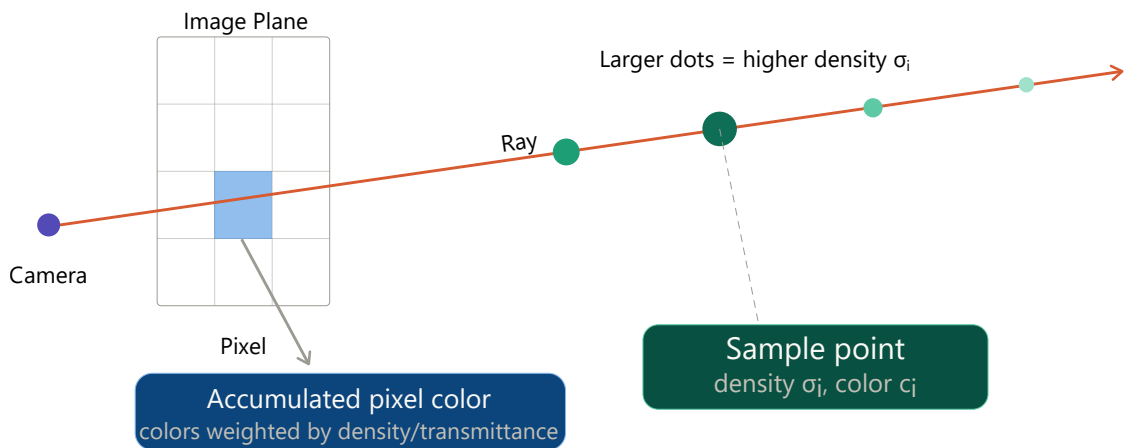
$$C(r) = \int_{t_n}^{t_f} T(t) \sigma(r(t)) c(r(t), d) dt, \quad T(t) = \exp \left(- \int_{t_n}^t \sigma(r(s)) ds \right). \quad (4)$$

O termo $T(t)$ é a transmitância: a fração de luz que sobrevive ao trajeto da câmera até a profundidade t sem ser bloqueada. Na prática, a integral é aproximada por uma soma sobre N amostras discretas $t_1 < \dots < t_N$ ao longo do raio:

$$\hat{C}(r) = \sum_{i=1}^N T_i \left(1 - e^{-\sigma_i \delta_i} \right) c_i, \quad T_i = \exp \left(- \sum_{j < i} \sigma_j \delta_j \right), \quad (5)$$

na qual $\delta_i = t_{i+1} - t_i$ é a distância entre amostras consecutivas. A leitura intuitiva é uma composição alfa de trás para frente: cada amostra contribui com sua cor proporcionalmente à sua opacidade local $(1 - e^{-\sigma_i \delta_i})$ e à luz que ainda chega até ela (T_i). A mesma soma ponderada, trocando c_i pela profundidade t_i , produz o mapa de profundidade esperado do raio, quantidade central para este trabalho: é essa profundidade renderizada que permite retroprojetar pixels para o espaço 3D no módulo de consistência multivista (Capítulo 4). A soma dos pesos, por sua vez, gera o mapa de acumulação, que indica quanto do raio atingiu superfície, em vez de fundo vazio. A Figura 2 esquematiza esse lançamento de raios e a composição ao longo deles.

Figura 2 – Esquema simplificado da renderização volumétrica em NeRF. Para cada pixel, um raio é lançado a partir da câmera e amostras 3D são avaliadas ao longo desse raio. A rede neural prediz cor e densidade em cada amostra, e a cor final do pixel é obtida por uma composição ponderada pelas densidades e transmitâncias acumuladas.

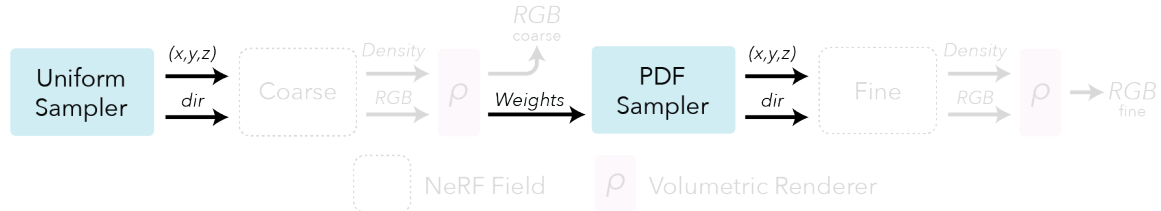


Fonte: Produção do autor.

Amostragem hierárquica. Avaliar a rede em centenas de pontos por raio é custoso, e a maior parte do espaço de uma cena é vazia. Por isso, o NeRF original adota uma estratégia em dois estágios: um amostrador uniforme distribui amostras ao longo de toda a faixa de profundidade e alimenta uma rede “grossa” (*coarse*); os pesos resultantes da composição definem uma distribuição de probabilidade ao longo do raio, da qual um segundo amostrador (*PDF sampler*) sorteia amostras concentradas nas regiões de maior peso (tipicamente perto das superfícies) para uma rede “fina” (*fine*) (Mildenhall et al., 2020; Nerfstudio Team, 2023). O efeito prático é gastar capacidade de rede onde há conteúdo, e não em espaço vazio. A Figura 3 ilustra os dois estágios de amostragem.

O pipeline modular do Nerfstudio. O Nerfstudio organiza esses componentes (geração de raios a partir das câmeras, amostradores, campo neural com codificação posicional e renderizadores de cor, profundidade e acumulação) como módulos intercambiáveis (Tancik et al., 2023), como ilustra a Figura 4. Essa modularidade é dire-

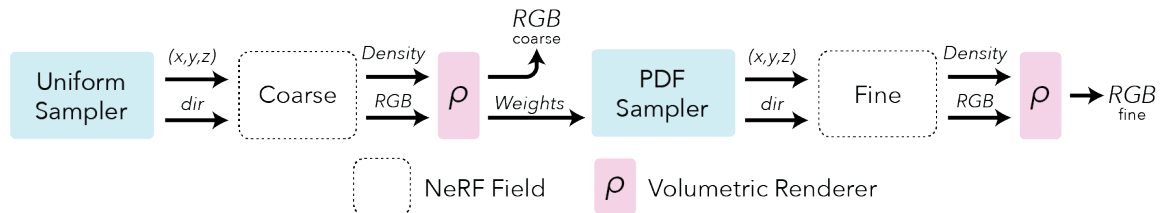
Figura 3 – Estratégia de amostragem hierárquica do NeRF na implementação do Nerfstudio: um amostrador uniforme cobre a faixa de profundidades e um amostrador por densidade de probabilidade (*PDF sampler*) concentra as amostras finais nas regiões próximas às superfícies.



Fonte: Documentação oficial do Nerfstudio (Nerfstudio Team, 2023).

tamente explorada neste trabalho: o método *nerfacto*, usado na etapa de reconstrução, troca componentes do NeRF original (por exemplo, o amostrador e a codificação) por variantes mais eficientes, mantendo a mesma interface de renderização diferenciável; e os renderizadores de profundidade e acumulação fornecem os sinais geométricos consumidos pelos módulos de edição propostos.

Figura 4 – Visão geral do *pipeline* NeRF como implementado no Nerfstudio: raios são gerados a partir das câmeras, amostrados em dois estágios, avaliados pelo campo neural e compostos pelos renderizadores.



Fonte: Documentação oficial do Nerfstudio (Nerfstudio Team, 2023).

O custo de renderização cresce com o número de pixels, o número de amostras por raio e o custo da avaliação da rede. Apesar disso, NeRFs são atraentes para este trabalho porque já estão integrados ao Nerfstudio e permitem otimização por gradientes de forma direta.

2.1.2 3D Gaussian Splatting

Uma alternativa explícita ao NeRF, também diferenciável, é o 3D Gaussian Splatting, que representa a cena como uma coleção de Gaussianas 3D com posição, escala, rotação, opacidade e cor dependente da vista (Kerbl et al., 2023). Cada Gaussiana é projetada no plano da imagem e renderizada por rasterização diferenciável com composição alfa. Em comparação com NeRFs tradicionais, essa representação tende a renderizar mais rapidamente e oferece uma estrutura explícita mais favorável a edições locais.

Embora a validação deste trabalho utilize o caminho *nerfacto* do Nerfstudio, a formulação proposta opera principalmente no nível da perda de edição por difusão. Por isso, os componentes que atuam sobre as previsões de ruído e os *scores* renderizados (direcionamento por TAG e STG, máscara auto-derivada, máscara de *Cross-Attention*, *source-blended localization* e âncoras de preservação) são conceitualmente aplicáveis tanto a NeRF quanto a 3D Gaussian Splatting. O módulo de *voxel cache* (Parte 2), por depender de mapas de profundidade e raios fornecidos pelo renderizador NeRF, é o componente menos diretamente transferível. Ambas as representações, contudo, dependem de poses de câmera conhecidas, normalmente obtidas por SfM, descrito a seguir.

2.1.3 Estrutura a partir de movimento

O fluxo de reconstrução depende de estimativas de câmera obtidas por SfM. Em geral, esse processo detecta e associa características entre imagens, estima poses de câmera e refina poses e pontos 3D por ajuste de feixes. O COLMAP é uma implementação amplamente utilizada desse *pipeline* (Schoenberger; Frahm, 2016). As poses e os parâmetros intrínsecos resultantes são usados pelo Nerfstudio para treinar a representação 3D inicial. A qualidade dessas poses importa além da reconstrução: como o módulo de consistência multivista da Parte 2 retroprojeta o sinal de edição de cada vista para o espaço 3D usando as câmeras estimadas, erros de pose propagam-se como desalinhamento na grade de *voxels* e podem atenuar o ganho de consistência pretendido.

Com a representação 3D definida, o próximo passo é explicar de onde vem o sinal de edição. Neste trabalho, esse sinal não é supervisionado por pares 3D editados, mas por um modelo de difusão 2D aplicado às vistas renderizadas, tema da próxima seção.

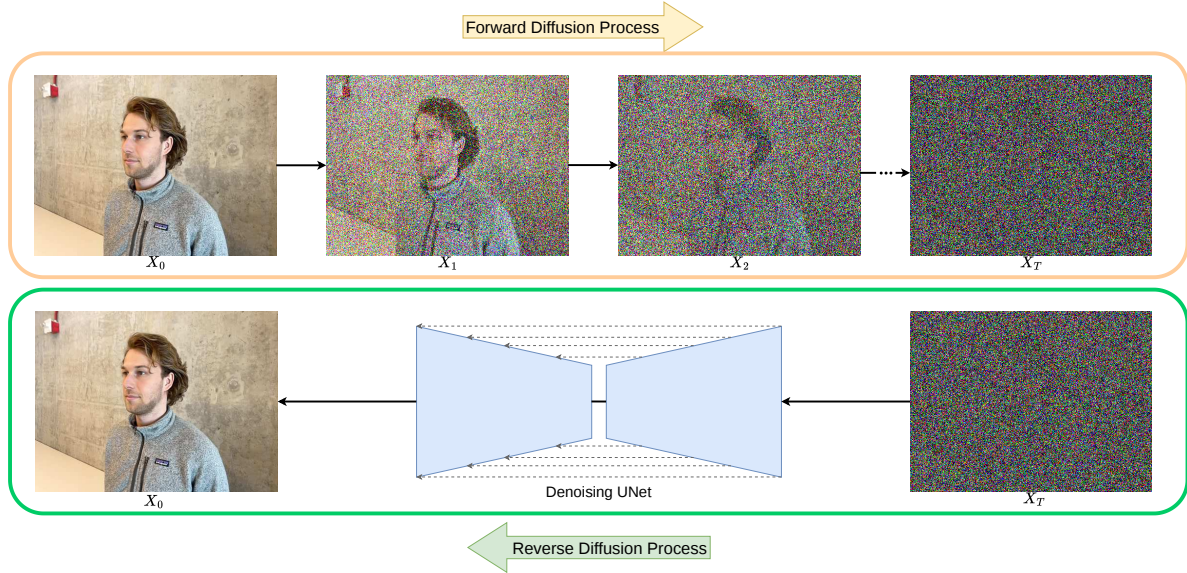
2.2 MODELOS DE DIFUSÃO E EDIÇÃO 2D

Modelos de difusão geram imagens aprendendo a inverter um processo de adição gradual de ruído (Ho; Jain; Abbeel, 2020). Em sua forma simplificada, uma amostra limpa x_0 é corrompida em um passo t por

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (6)$$

na qual ϵ é um ruído gaussiano e $\bar{\alpha}_t \in [0, 1]$ é o produto acumulado dos coeficientes do escalonamento de ruído até o passo t , que controla quanto do sinal original sobrevive: com t pequeno, $\bar{\alpha}_t \approx 1$ e x_t permanece próxima de x_0 ; com t grande, $\bar{\alpha}_t \rightarrow 0$ e x_t aproxima-se de ruído puro. O processo direto (adição de ruído) e o reverso (remoção) são ilustrados na Figura 5.

Figura 5 – Esquema do processo direto e reverso em modelos de difusão. No processo direto, ruído é adicionado progressivamente a uma amostra limpa x_0 até aproximá-la de ruído puro x_T . No processo reverso, uma rede de remoção de ruído estima como transformar uma amostra ruidosa em uma amostra limpa.



Fonte: Produção do autor.

O modelo é treinado para prever o ruído $\epsilon_\theta(x_t, t, c)$ condicionado por um sinal c , como um texto.

2.2.1 Classifier-Free Guidance

Classifier-Free Guidance (CFG) combina previsões condicionais e incondicionais para aumentar a aderência ao *prompt* (Ho; Salimans, 2022):

$$\epsilon_{\text{cfg}} = \epsilon_{\emptyset} + s(\epsilon_c - \epsilon_{\emptyset}), \quad (7)$$

na qual ϵ_c é a previsão de ruído condicionada ao sinal c , $\epsilon_{\emptyset}$ é a previsão incondicional (o subscrito $\emptyset$ denota condição vazia) e s controla a força do *guidance*. Valores altos de s podem produzir edições mais intensas, mas também aumentam o risco de saturação, perda de fidelidade ou convergência para um modo visual excessivamente estereotipado.

2.2.2 InstructPix2Pix

InstructPix2Pix (IP2P) é um modelo de edição 2D condicionado por uma imagem fonte e uma instrução textual (Brooks; Holynski; Efros, 2023). Diferentemente de um modelo texto-para-imagem puro, o IP2P recebe o latente ruidoso concatenado ao latente da imagem fonte. Na formulação usada pelo DreamCatalyst, a previsão final

segue um CFG de três termos:

$$\epsilon_{\text{cfg}} = \epsilon_{\text{uncond}} + s_{\text{text}}(\epsilon_{\text{text}} - \epsilon_{\text{image}}) + s_{\text{image}}(\epsilon_{\text{image}} - \epsilon_{\text{uncond}}), \quad (8)$$

na qual ϵ_{text} , ϵ_{image} e ϵ_{uncond} são as predições de ruído condicionadas ao texto, à imagem fonte e sem condicionamento, respectivamente; a escala s_{text} controla a aderência à instrução textual e s_{image} a fidelidade à imagem fonte. Os valores dessas escalas são fixados no protocolo experimental (Capítulo 6). Essa estrutura é adequada para edição 3D porque cada vista renderizada da cena pode ser tratada como a imagem fonte do IP2P, como esquematiza a Figura 6.

Na edição 3D via DDS, contudo, os dois ramos que compõem o gradiente (Seção 2.3) instanciam esse CFG de forma *assimétrica*. O ramo alvo, associado à vista renderizada corrente, recebe a instrução de edição e usa os três termos completos:

$$\epsilon_{\text{cfg}}^{\text{tgt}} = \epsilon_{\text{uncond}} + s_{\text{text}}(\epsilon_{\text{text}} - \epsilon_{\text{image}}) + s_{\text{image}}(\epsilon_{\text{image}} - \epsilon_{\text{uncond}}). \quad (9)$$

O ramo fonte, associado à vista original da mesma câmera, *omite o termo textual* e conserva apenas o condicionamento de imagem, o que equivale a fazer $s_{\text{text}} = 0$:

$$\epsilon_{\text{cfg}}^{\text{src}} = \epsilon_{\text{uncond}} + s_{\text{image}}(\epsilon_{\text{image}} - \epsilon_{\text{uncond}}). \quad (10)$$

Cada ramo avalia os termos sobre a sua própria vista ruidosa (a renderização corrente, no alvo; a vista fonte original, na fonte), corrompida com o mesmo ruído e no mesmo passo t .

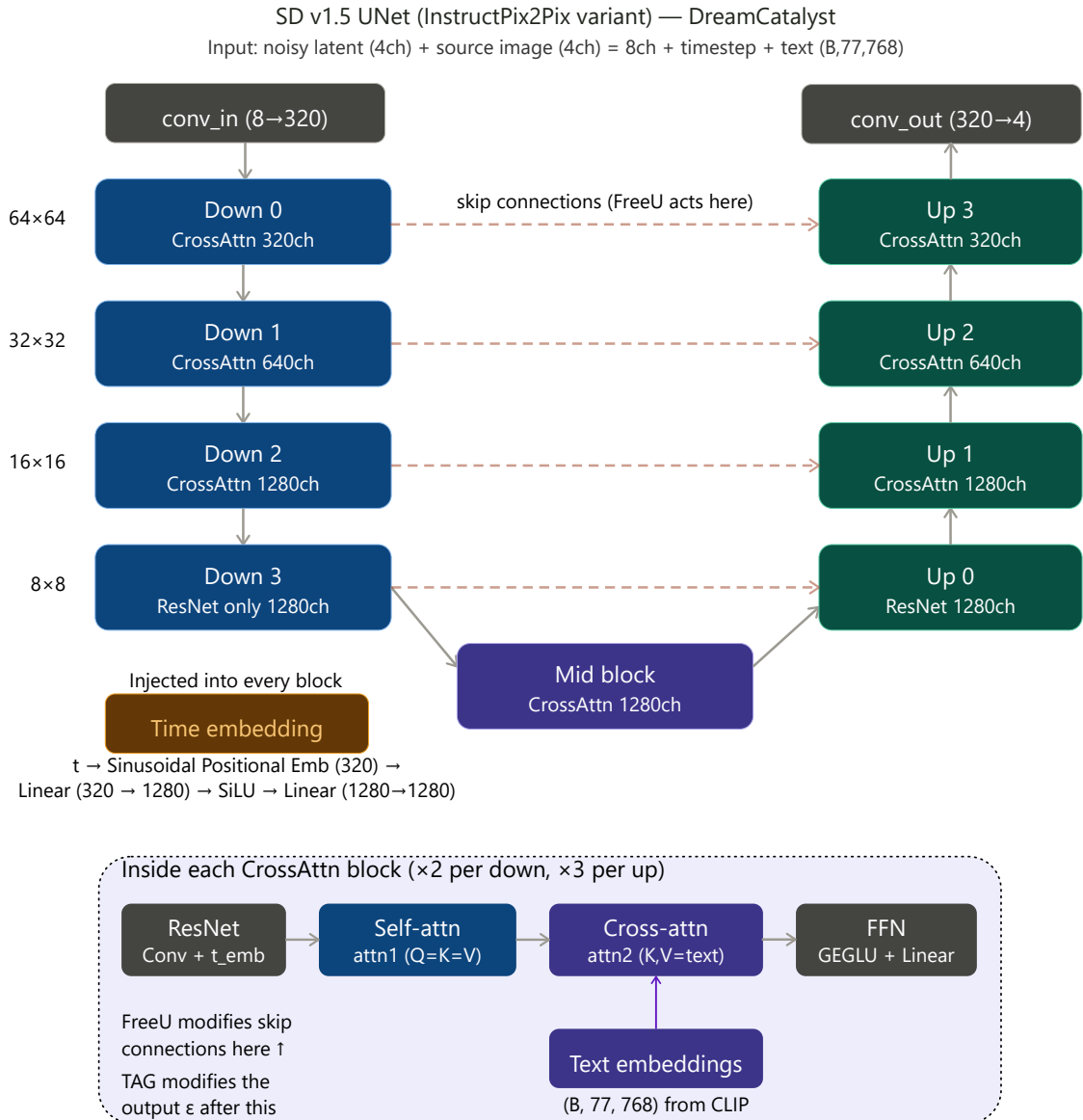
A assimetria decorre do papel de cada ramo no DDS. Como o IP2P é condicionado pela imagem fonte, esse condicionamento já ancora os dois ramos à aparência original da cena; zerar o termo textual reduz o IP2P a uma reconstrução fiel dessa imagem, sem aplicar qualquer instrução. O ramo fonte funciona, então, como uma *referência de reconstrução* da vista original, enquanto o ramo alvo acrescenta a direção textual da edição. A diferença $\epsilon_{\text{cfg}}^{\text{tgt}} - \epsilon_{\text{cfg}}^{\text{src}}$ cancela as componentes comuns e alheias à edição (o termo incondicional e a reconstrução guiada pela imagem) e isola o efeito da instrução textual. Se o ramo fonte também recebesse o termo textual, a referência deixaria de ser uma reconstrução fiel da cena original e se tornaria ela própria uma edição, contaminando a direção estimada e enfraquecendo a preservação que o DDS busca garantir.

A partir desse ponto, a questão deixa de ser apenas gerar uma boa imagem 2D. O IP2P fornece predições de ruído para uma vista renderizada; a edição 3D precisa transformar essas predições em um gradiente estável para os parâmetros da cena. É essa ponte entre difusão 2D e otimização 3D que motiva o SDS e o DDS.

2.2.3 Blocos internos do UNet

Como o IP2P é construído sobre uma arquitetura do tipo UNet, vale localizar onde STG e *Cross-Attention* atuam. Em alto nível, o codificador reduz a resolução espa-

Figura 6 – Esquema da arquitetura de remoção de ruído usada pelo InstructPix2Pix. O latente ruidoso da vista renderizada é concatenado ao latente da imagem fonte, enquanto a instrução textual condiciona os blocos de *Cross-Attention* do UNet.



Fonte: Produção do autor.

cial enquanto aumenta a capacidade semântica, o bloco central (*mid-block*) concentra informação global e o decodificador reconstrói a resolução do latente; blocos residuais mantêm a estabilidade da otimização e blocos com *Cross-Attention* conectam *tokens* textuais às posições espaciais do latente. O STG (*Spatiotemporal Skip Guidance*), detalhado no Capítulo 4, reforça o sinal de edição perturbando seletivamente esses blocos internos.

Os *Down blocks* e *Up blocks* formam a espinha dorsal codificador–decodificador, e os blocos com *Cross-Attention* inserem o condicionamento textual ao longo desse fluxo, tanto no codificador quanto no bloco central e no decodificador.

No caminho do decodificador, o *Cross-Attention Up block* (Figura 7) atua enquanto a resolução latente é recuperada. Esse bloco é especialmente relevante para este trabalho, pois os mapas de atenção capturados nos *up-blocks* preservam melhor a relação entre *tokens* textuais e regiões espaciais da imagem renderizada.

Neste trabalho, os mapas de *Cross-Attention* são importantes por dois motivos. Primeiro, eles explicam como o IP2P relaciona os *tokens* do *prompt* alvo, denotado t_{tgt} (e t_{src} para o *prompt* fonte; o subscrito textual distingue-os do passo de difusão t), com regiões do latente. Segundo, eles servem como base para a máscara semântica M_{attn} , derivada desses mapas de atenção e usada posteriormente para modular a *máscara auto-derivada* (obtida da diferença entre as previsões de ruído fonte e alvo, detalhada no Capítulo 4) e assim reduzir edições fora da região semanticamente relevante.

2.3 SCORE DISTILLATION E EDIÇÃO 3D

Score Distillation Sampling (SDS) utiliza um modelo de difusão 2D para otimizar uma representação que não é diretamente uma imagem, como uma malha, um NeRF ou uma nuvem de Gaussianas (Poole et al., 2023). Se $x = g(\theta)$ é a imagem renderizada por uma representação 3D com parâmetros θ , uma forma comum do gradiente de SDS é

$$\nabla_{\theta} \mathcal{L}_{SDS} = w(t) (\epsilon_{\theta}(x_t, t, c) - \epsilon) \frac{\partial x}{\partial \theta}, \quad (11)$$

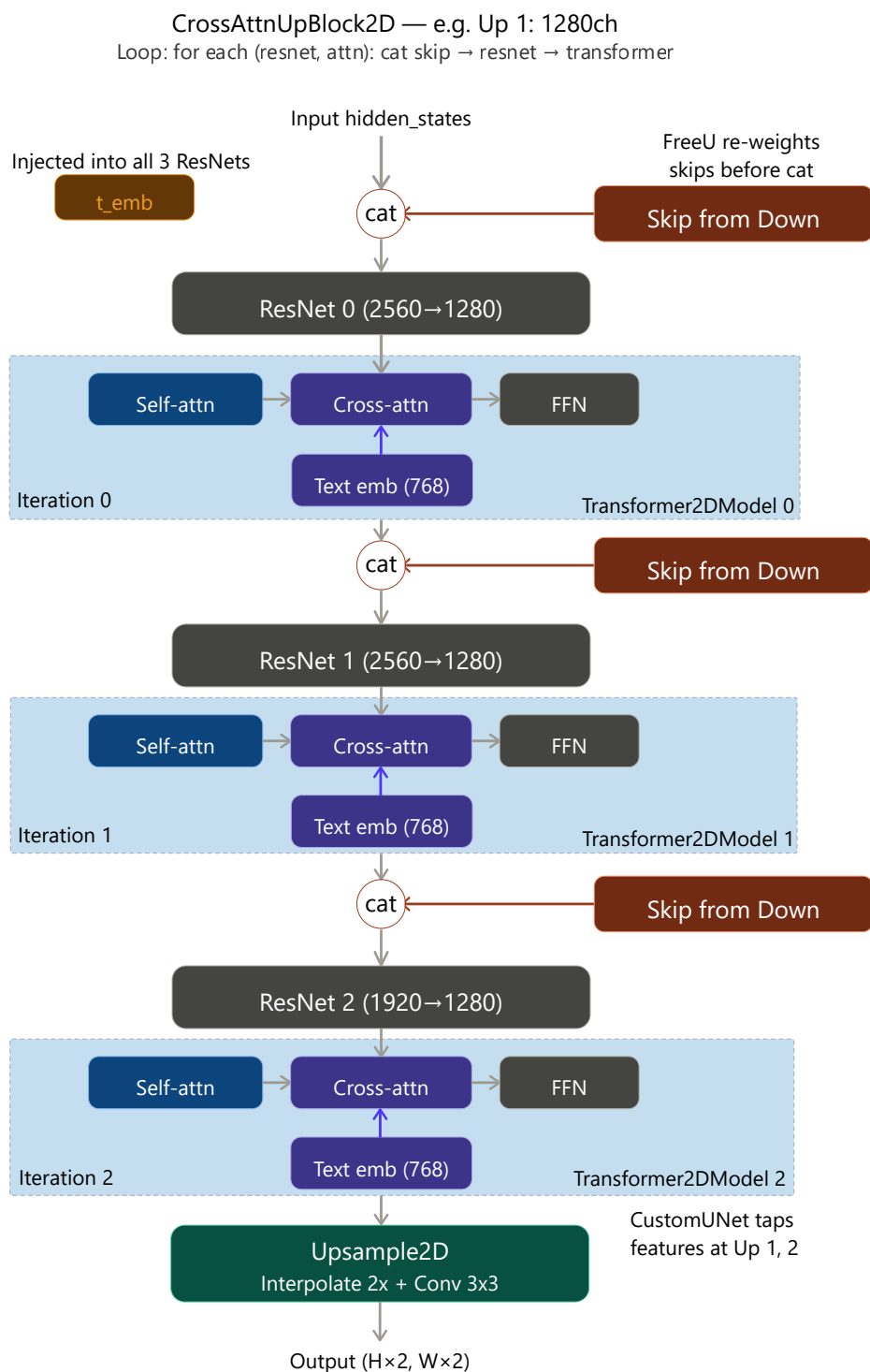
na qual $w(t)$ é uma função de ponderação dependente do passo de difusão t , que equilibra a contribuição dos diferentes níveis de ruído ao longo da otimização; c é o condicionamento textual; e ϵ é o ruído usado na corrupção direta. Apesar de eficaz para geração 3D, o SDS pode apresentar alta variância e tendência a artefatos.

Delta Denoising Score (DDS) substitui a comparação com o ruído aleatório por uma diferença entre os *scores* do alvo e da fonte (Hertz et al., 2023):

$$\nabla_{\theta} \mathcal{L}_{DDS} = w(t) (\epsilon_{\theta}^{tgt} - \epsilon_{\theta}^{src}) \frac{\partial x}{\partial \theta}, \quad (12)$$

na qual ϵ_{θ}^{tgt} e ϵ_{θ}^{src} são as previsões de ruído dos ramos alvo e fonte, e $w(t)$ é o mesmo peso por passo da equação de SDS. Cada ramo é avaliado sobre a sua própria vista, corrompida com o mesmo ruído e no mesmo passo t : o ramo alvo processa a vista renderizada corrente sob a instrução de edição (t_{tgt}); o ramo fonte processa a vista fonte original, instanciada com IP2P como uma reconstrução fiel dessa vista, com condicionamento de imagem apenas (Seção 2.2.2). Essa diferença representa a direção de edição: o quanto a vista renderizada deve se afastar da reconstrução fiel da fonte na direção indicada pela instrução alvo. O DreamCatalyst utiliza essa ideia como base para editar representações 3D treinadas previamente.

Figura 7 – *Cross-Attention Up block*. No caminho do decodificador, a *Cross-Attention* ajuda a transformar a instrução textual em alterações espacialmente localizadas enquanto a resolução latente é recuperada.



Fonte: Produção do autor.

Entretanto, o DDS ainda é um sinal por vista. Para que ele seja útil em uma cena 3D, não basta saber a direção de edição; também é necessário decidir onde essa direção deve atuar e onde a cena deve permanecer ancorada à reconstrução original.

2.4 LOCALIZAÇÃO ESPACIAL E *CROSS-ATTENTION*

Uma dificuldade central da edição por difusão é localizar onde a instrução deve atuar. Métodos como o DiffEdit constroem máscaras a partir de diferenças entre predições de ruído (Couairon et al., 2023). Outros trabalhos exploram mapas de *Cross-Attention*, pois a atenção entre *tokens* textuais e posições espaciais carrega informação sobre a região associada a cada palavra (Hertz et al., 2022; Tang et al., 2022). O LEDITS++ combina máscaras de atenção e ruído para controlar melhor edições locais (Brack et al., 2024).

Neste trabalho, essas ideias são adaptadas para o contexto de DDS em 3D. A diferença entre predições de ruído fonte e alvo indica onde o modelo de difusão enxerga mudança relevante, enquanto a *Cross-Attention* fornece um *prior* semântico associado aos *tokens* do texto alvo t_{tgt} . As duas fontes são combinadas em uma única máscara de localização, cujo detalhamento é apresentado no Capítulo 4.

2.5 CONSISTÊNCIA ENTRE VISTAS

Quando uma cena 3D é editada por vistas renderizadas independentemente, não há garantia de que duas câmeras distintas recebam máscaras ou gradientes compatíveis. O modo de falha concreto é uma mesma região física receber decisões opostas conforme o ângulo: um braço pintado em uma vista e preservado em outra, por exemplo. Como a otimização 3D integra gradientes de muitas vistas ao longo de milhares de iterações, essas decisões incompatíveis se acumulam na representação.

Trabalhos recentes abordam esse problema elevando sinais *semânticos* 2D para representações 3D (Zhi et al., 2021; Siddiqui et al., 2023) ou impondo consistência entre vistas na edição de Gaussianas 3D diretamente sobre o sinal de edição (Wen et al., 2025). Na mesma linha, este trabalho investiga um *voxel cache* que agrega, em uma grade 3D, a magnitude de edição observada por cada vista, de modo que a evidência de uma câmera possa influenciar as demais. Sua formulação e avaliação, incluindo o regime em que ele ajuda e o regime em que apenas amplifica sinais consistentes mas indesejados, são tratadas no Capítulo 4 (formulação) e na Parte 2 do Capítulo 6 (resultados).

Estabelecidos esses fundamentos (representação 3D diferenciável, sinal de difusão, destilação de *score* e localização), o Capítulo 3 formaliza precisamente o problema de edição 3D tratado e os requisitos que a solução proposta deve atender.

3 DESCRIÇÃO DO PROBLEMA E REQUISITOS TÉCNICOS

Este capítulo descreve com mais precisão o problema tratado neste trabalho. O trabalho não busca criar um novo modelo de difusão nem um novo método de reconstrução 3D. O foco é a etapa de edição: dada uma cena 3D já reconstruída, como aplicar um comando textual de forma localizada, preservadora e coerente entre vistas.

3.1 FLUXO DE REFERÊNCIA

O fluxo de referência utilizado neste trabalho possui quatro etapas:

1. processamento das imagens e estimação de câmeras por COLMAP;
2. treinamento de uma representação 3D em Nerfstudio, com *nerfacto* (NeRF) nos experimentos deste trabalho e *splatfacto* (3D Gaussian Splatting) fora do escopo experimental;
3. edição por DreamCatalyst, usando DDS e IP2P sobre vistas renderizadas;
4. avaliação visual e quantitativa das vistas editadas.

A terceira etapa é o objeto principal deste trabalho. A cena 3D inicial é mantida como ponto de partida, e a otimização modifica seus parâmetros para que as vistas renderizadas passem a satisfazer t_{tgt} , isto é, o texto que descreve a edição desejada.

3.2 FORMULAÇÃO DO PROBLEMA

Considere uma representação 3D com parâmetros θ e um renderizador diferenciável g . Para uma câmera v , a imagem renderizada é

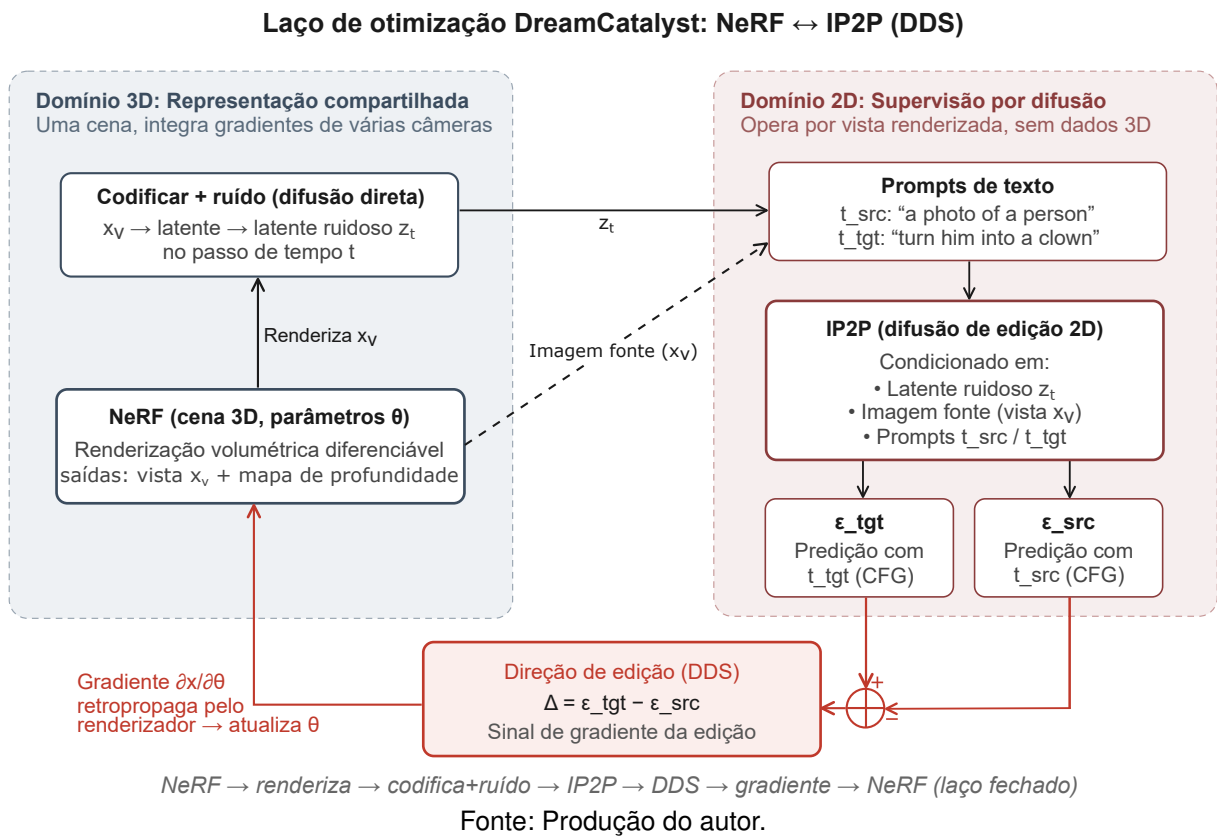
$$x_v = g(\theta, v). \quad (13)$$

O usuário fornece t_{src} , que descreve a cena original, e t_{tgt} , que descreve a edição desejada. O objetivo é otimizar θ para que x_v siga t_{tgt} nas regiões relevantes, mas preserve estrutura, identidade e fundo compatíveis com t_{src} .

A Figura 8 resume o laço de otimização do DreamCatalyst base (sem os módulos propostos neste trabalho), que instancia a terceira etapa do fluxo de referência. A cada passo, o NeRF renderiza uma vista x_v ; essa vista é codificada e ruidificada em um latente z_t ; o IP2P, condicionado pela vista fonte e pelos *prompts* t_{src} e t_{tgt} , produz duas previsões de ruído cuja diferença $\epsilon_{tgt} - \epsilon_{src}$ é o sinal de edição DDS; e esse sinal retropropaga pelo renderizador diferenciável para atualizar θ . O ponto central é o acoplamento entre os dois domínios: o IP2P supervisiona em 2D, vista a vista, enquanto o NeRF integra esses gradientes em uma representação 3D compartilhada ao longo

de muitas câmeras e iterações. Não há dados 3D editados; há apenas a orientação do modelo de difusão 2D sobre as vistas renderizadas.

Figura 8 – Laço de otimização do DreamCatalyst base: interação entre o NeRF (domínio 3D, parâmetros θ) e o IP2P (domínio 2D, supervisão por difusão). O NeRF renderiza uma vista, que é codificada e ruidificada em z_t ; o IP2P, condicionado pela vista fonte e pelos *prompts* t_{src}/t_{tgt} , gera as previsões ϵ_{tgt} e ϵ_{src} ; a diferença entre elas (DDS) retropropaga pelo renderizador diferenciável e atualiza θ , fechando o laço. A supervisão é 2D e por vista, mas é integrada em uma única representação 3D.



O problema é difícil porque o modelo de difusão 2D avalia uma vista de cada vez. A direção de edição calculada para uma câmera, portanto, não tem acesso direto às máscaras ou decisões semânticas obtidas para outras câmeras. Em uma cena com várias vistas, essa independência pode produzir edições parcialmente incompatíveis em diferentes partes do mesmo objeto 3D.

Essa observação deve ser entendida como uma fragilidade do paradigma de supervisão por vistas, não como uma afirmação de que o DreamCatalyst original falha em todas as cenas. No escopo deste trabalho, a inconsistência entre vistas foi observada principalmente quando sinais de *guidance* mais fortes passaram a amplificar decisões locais distintas entre câmeras.

3.3 LIMITAÇÕES OBSERVADAS E RISCOS DO FLUXO DE REFERÊNCIA

O DreamCatalyst fornece uma base forte, mas o fluxo de referência apresenta limitações e riscos relevantes para este projeto:

- **Localização insuficiente:** o gradiente de DDS pode afetar regiões fora do objeto ou atributo que o usuário deseja editar;
- **Deriva de fundo:** superfícies que deveriam permanecer estáveis, como paredes, pisos e texturas de fundo, podem ser suavizadas ou alteradas durante a edição;
- **Risco de inconsistência entre vistas:** por depender de sinais 2D calculados por vista, a mesma região física pode receber decisões de edição diferentes ao ser observada por ângulos distintos, especialmente quando o *guidance* é amplificado;
- **Conflito entre força de edição e preservação:** aumentar o *guidance* textual tende a fortalecer a mudança semântica, mas também pode degradar identidade, geometria ou aparência original;
- **Dependência da cena e do *prompt*:** edições sutis, como alterar uma face, e edições estruturais, como transformar uma pessoa em um personagem com armadura, exigem equilíbrios diferentes entre edição e preservação.

3.4 REQUISITOS TÉCNICOS

A solução proposta deve atender aos seguintes requisitos:

1. **Compatibilidade com o *pipeline* existente:** a implementação deve operar dentro do fluxo Nerfstudio/DreamCatalyst, sem exigir treinamento de um novo modelo generativo 2D;
2. **Modularidade:** cada componente novo deve poder ser desligado para reproduzir o DreamCatalyst base;
3. **Generalidade:** o método não deve depender de máscaras manuais, segmentação externa por cena ou seleção manual de palavras-chave;
4. **Localização:** a direção de edição deve ser concentrada em regiões semanticamente relevantes;
5. **Preservação:** regiões fora da máscara devem ser ancoradas à aparência original;
6. **Consistência entre vistas:** sempre que possível, evidências de uma vista devem ser reutilizadas por outras vistas que observam a mesma região 3D;

7. **Avaliabilidade:** a solução deve produzir imagens, máscaras, métricas e diagnósticos suficientes para análise quantitativa e qualitativa.

3.5 ESCOPO

O escopo deste trabalho está restrito à etapa de edição de cenas já reconstruídas. A qualidade da reconstrução inicial continua sendo uma variável importante, mas não é o foco do trabalho. Da mesma forma, o refinamento posterior à edição não é tratado como contribuição principal. As extensões propostas atuam durante a otimização DDS, isto é, no momento em que o modelo de difusão fornece o gradiente usado para atualizar a representação 3D.

3.6 CRITÉRIOS DE AVALIAÇÃO

Os resultados são avaliados por critérios visuais e por métricas automáticas. As métricas incluem similaridade direcional em Contrastive Language-Image Pretraining (CLIP) para medir aderência semântica à edição, Learned Perceptual Image Patch Similarity (LPIPS) e Structural Similarity Index Measure (SSIM) para medir preservação visual, similaridade imagem-imagem em CLIP e uma métrica de consistência entre vistas; suas definições são detalhadas no Capítulo 5. A interpretação dessas métricas é feita com cautela, pois medidas automáticas podem não capturar artefatos localizados, como assimetrias entre braços ou pequenas falhas em regiões semanticamente importantes.

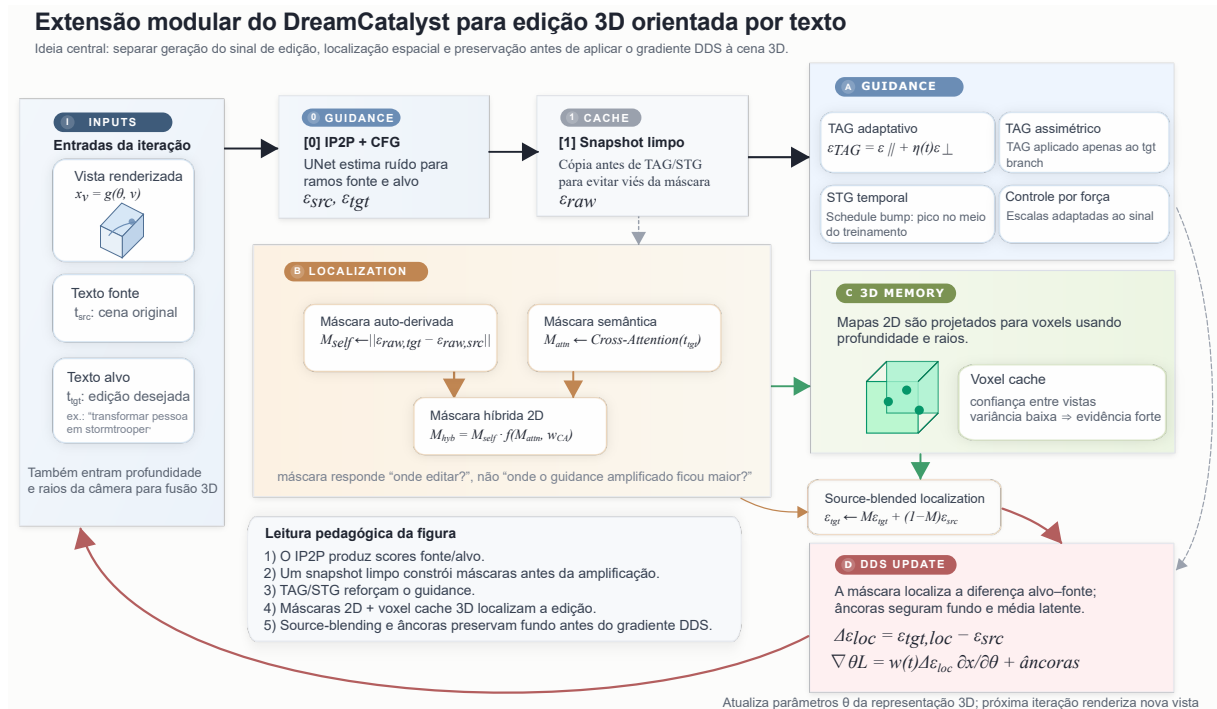
4 SOLUÇÃO PROPOSTA E METODOLOGIA UTILIZADA

Este capítulo apresenta a solução proposta. A estratégia geral consiste em manter o DreamCatalyst como método base e inserir módulos adicionais entre a predição de ruído do IP2P e o gradiente DDS aplicado à representação 3D. Esses módulos atuam em três frentes: aumentar a força da edição (*guidance*), localizar a região da edição e preservar as regiões que não devem ser alteradas.

4.1 VISÃO GERAL

A Figura 9 oferece uma leitura pedagógica desses módulos, retomando o laço base de otimização da Figura 8 (Capítulo 3) e indicando onde cada módulo proposto se encaixa. A ideia central é separar quatro decisões que, à primeira vista, poderiam parecer uma única etapa: gerar o sinal de edição com IP2P, construir uma máscara limpa de localização, reforçar o *guidance* com TAG/STG e preservar a cena antes de aplicar o gradiente DDS à representação 3D.

Figura 9 – Visão pedagógica da extensão modular do DreamCatalyst. O fluxo separa geração do sinal de edição, localização espacial, um módulo exploratório de memória multivista por *voxel cache* (opcional, avaliado na Parte 2) e preservação antes da atualização DDS da cena 3D.

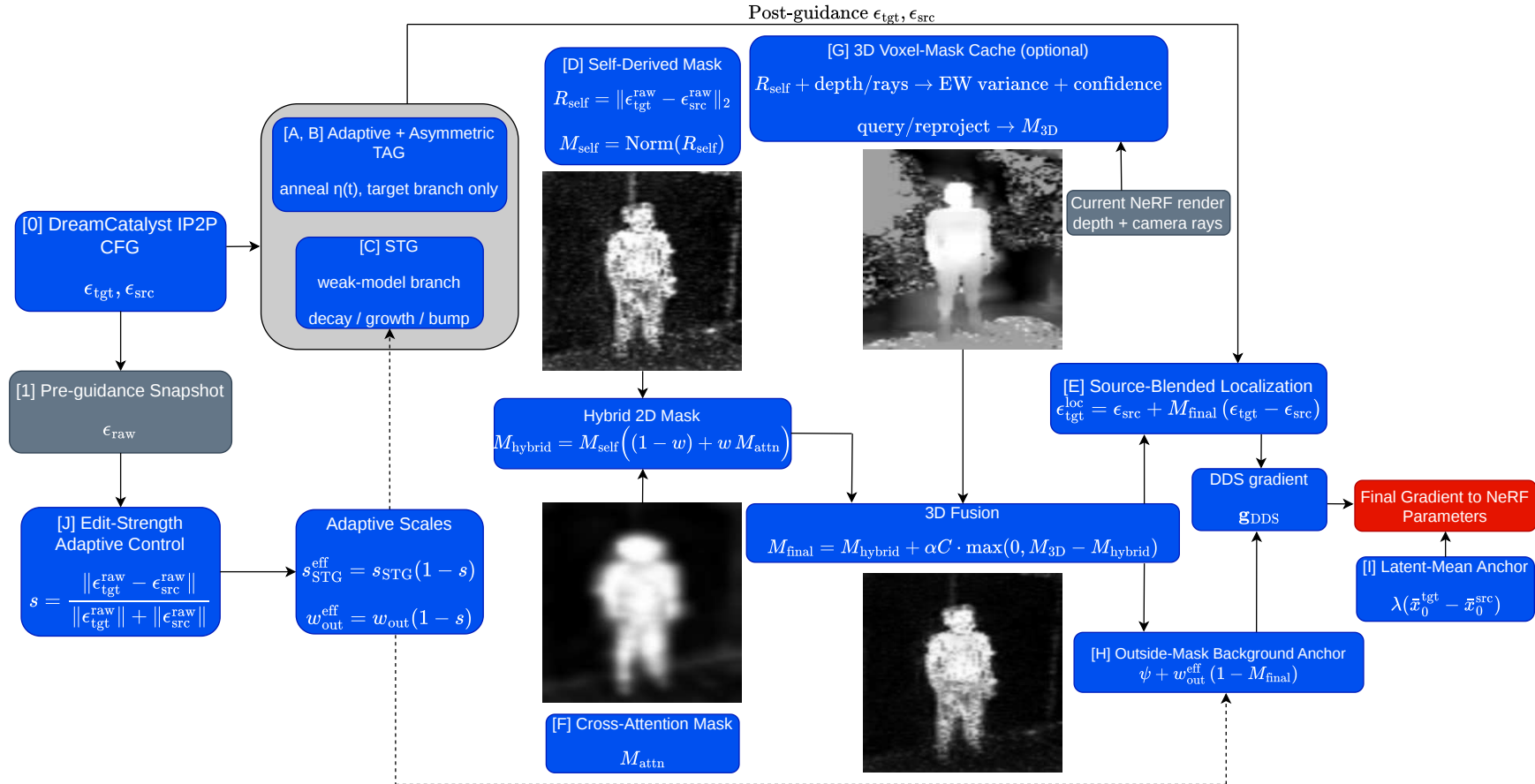


Fonte: Produção do autor.

A Figura 10 complementa essa leitura com o diagrama técnico dos módulos. A etapa inicial é o CFG do IP2P, que produz predições de ruído para os ramos fonte e alvo. Em seguida, uma cópia limpa dessas predições é armazenada. Essa cópia,

denominada ϵ^{raw} , serve para construir máscaras de relevância antes que TAG, STG ou outros mecanismos alterem o vetor de *guidance*. Só então os mecanismos de *guidance* modificam as previsões de ruído, a máscara final localiza a diferença entre alvo e fonte e o gradiente DDS é enviado à representação 3D.

Figura 10 – Visão geral dos módulos de *guidance*, localização e preservação adicionados ao DreamCatalyst.



Fonte: Produção do autor.

Algoritmo 1: Fluxo geral da edição 3D proposta.**Entrada:** Cena 3D reconstruída θ , câmera v , *prompts* t_{src} e t_{tgt} .**Saída:** Cena 3D atualizada por um passo de otimização.

- 1 Renderizar a vista atual $x_v = g(\theta, v)$
- 2 Enviar x_v , t_{src} e t_{tgt} ao IP2P e obter $\epsilon_{\text{src}}^{\text{raw}}$ e $\epsilon_{\text{tgt}}^{\text{raw}}$ logo após o CFG
- 3 Construir M_{self} pela diferença entre previsões de ruído e M_{attn} por *Cross-Attention*
- 4 Combinar as máscaras em M_{hybrid} e, se habilitado, consultar o *voxel cache* para obter M_{final}
- 5 Aplicar TAG, STG e demais módulos de *guidance* nas previsões de ruído
- 6 Misturar fonte e alvo por *source-blended localization* usando M_{final}
- 7 Calcular o gradiente DDS, somar âncoras de preservação e atualizar θ
- 8 Atualizar o *voxel cache* com a máscara observada, profundidade e raios da câmera

4.2 PREDIÇÕES LIMPAS PARA LOCALIZAÇÃO

Uma decisão metodológica importante é separar a construção da máscara da aplicação do *guidance*. As máscaras são calculadas a partir de ϵ^{raw} , isto é, das previsões obtidas após o CFG do IP2P, mas antes de TAG e STG. Essa escolha evita que a máscara herde artificialmente artefatos criados pelos próprios mecanismos de amplificação. A máscara deve responder à pergunta “onde a edição parece relevante?” e não à pergunta “onde o *guidance* amplificado ficou maior?”.

4.3 DIRECIONAMENTO ADAPTATIVO

4.3.1 TAG adaptativo e assimétrico

TAG decompõe uma previsão de ruído em componentes radial e tangencial em relação ao latente ruidoso z_t (Cho et al., 2025). Definindo

$$v = \frac{z_t}{\|z_t\|}, \quad (14)$$

a componente radial (paralela a v , denotada $\epsilon_{\parallel}$) é a projeção escalar do ruído sobre v multiplicada pelo próprio v :

$$\epsilon_{\parallel} = \left(\sum_i \epsilon_i v_i \right) v, \quad (15)$$

e a componente tangencial é

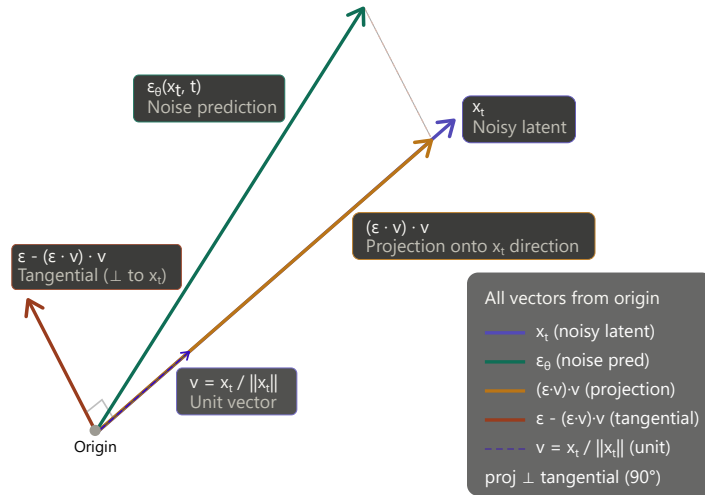
$$\epsilon_{\perp} = \epsilon - \epsilon_{\parallel}. \quad (16)$$

O ruído guiado por TAG é então

$$\epsilon_{\text{TAG}} = \epsilon_{\parallel} + \eta \epsilon_{\perp}, \quad (17)$$

na qual $\eta \geq 0$ é o fator de escala aplicado à componente tangencial: $\eta = 1$ recupera a predição original e $\eta > 1$ reforça a direção tangencial do *score* sem inflar a componente radial (a norma alinhada ao latente ruidoso). A Figura 11 apresenta a interpretação geométrica dessa decomposição.

Figura 11 – Interpretação geométrica de TAG. A componente radial controla a direção alinhada ao latente ruidoso, enquanto a componente tangencial altera a direção efetiva do *score* sem simplesmente aumentar toda a norma da predição.



Fonte: Produção do autor.

Neste trabalho, TAG é usado de forma adaptativa e assimétrica. A versão adaptativa adota um $\eta(t)$ maior em *timesteps* mais ruidosos e o aproxima de 1 em *timesteps* mais limpos. A versão assimétrica aplica TAG ao ramo alvo, mas mantém o ramo fonte como âncora estrutural.

4.3.2 STG com escalonamento temporal

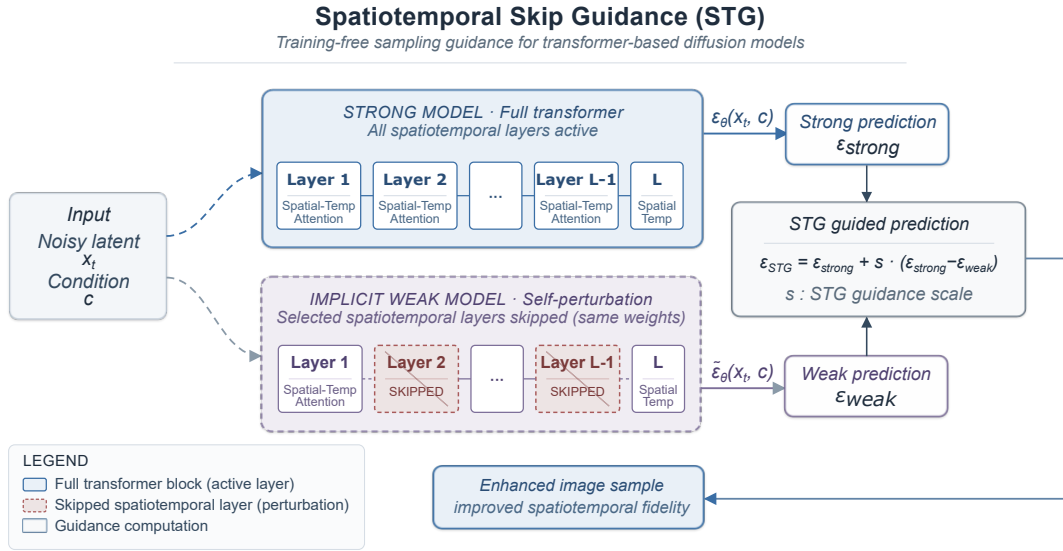
STG realiza uma segunda passagem fraca pelo modelo, na qual camadas de atenção são parcialmente degradadas, e amplifica a diferença entre a predição completa e a fraca (Hyung et al., 2025):

$$\epsilon_{\text{STG}} = \epsilon_{\text{full}} + s_{\text{STG}} (\epsilon_{\text{full}} - \epsilon_{\text{weak}}), \quad (18)$$

na qual ϵ_{full} é a predição completa, ϵ_{weak} a predição da passagem enfraquecida e $s_{\text{STG}} \geq 0$ a escala que controla quanto a diferença entre as duas é amplificada. A Figura 12 resume esse mecanismo de duas passagens.

Como STG pode reforçar estados parciais diferentes em vistas distintas, este trabalho emprega uma janela temporal controlada. A versão principal usa um escalonamento em forma de pico, no qual STG cresce no meio da otimização e decai antes do final. A motivação é permitir que STG auxilie na formação de estrutura durante a fase

Figura 12 – Mecanismo original do STG (*Spatiotemporal Skip Guidance*), de Hyung et al. (Hyung et al., 2025), proposto para modelos de difusão baseados em *transformers* (originalmente, difusão de vídeo): uma predição completa é comparada com uma predição fraca, obtida ao pular camadas internas do modelo, e a diferença entre os dois ramos reforça direções estruturais do *score*. Neste trabalho, o mesmo princípio de *skip-guidance* é adaptado ao UNet do InstructPix2Pix, perturbando suas camadas de atenção.



Fonte: Produção do autor.

intermediária, sem congelar inconsistências entre vistas no final. A Figura 13 compara os padrões de escalonamento disponíveis (*decay*, *growth* e *bump*).

4.4 MÁSCARA AUTO-DERIVADA

A máscara auto-derivada usa a magnitude da diferença entre as predições de ruído alvo e fonte:

$$R = \left\| \epsilon_{\text{tgt}}^{\text{raw}} - \epsilon_{\text{src}}^{\text{raw}} \right\|_2. \quad (19)$$

O mapa R é normalizado por percentis, suavizado e, opcionalmente, aguçado por um expoente γ . O resultado é uma máscara suave $M_{\text{self}} \in [0, 1]$. A Figura 14 ilustra uma vista renderizada e as máscaras usadas durante a edição.

4.5 MÁSCARA SEMÂNTICA POR *CROSS-ATTENTION*

A máscara por *Cross-Attention* explora mapas internos da arquitetura UNet do IP2P. Os *tokens* usados são derivados automaticamente de t_{tgt} , removendo-se palavras comuns e palavras presentes em t_{src} . Por exemplo, em uma instrução como “transformar uma pessoa em stormtrooper”, o termo semanticamente mais informativo

Figura 13 – Padrões de escalonamento (*schedule shapes*) usados para controlar a escala relativa de STG. O modo *bump* concentra a amplificação na fase intermediária e libera a fase final para refinamento sem congelar estados parciais dependentes da vista.

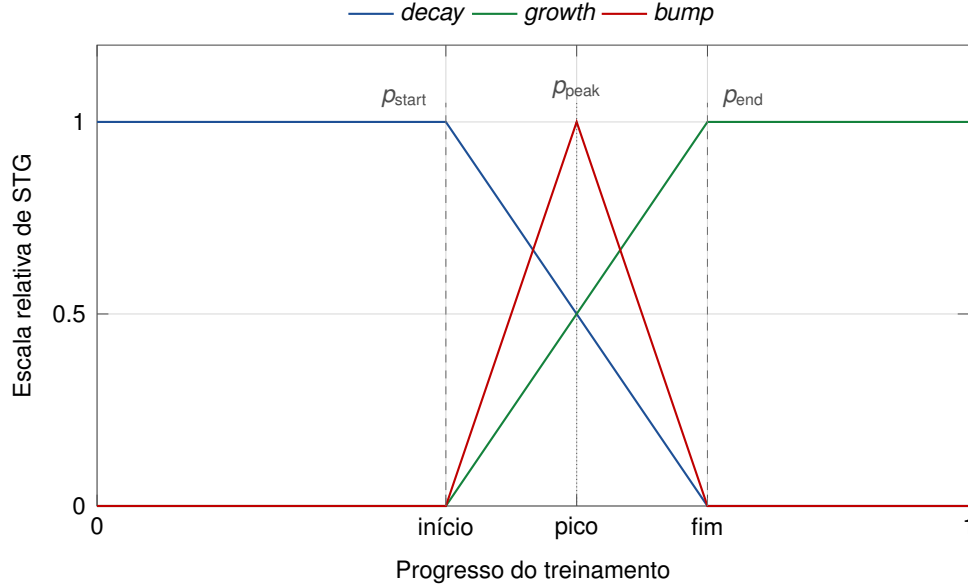
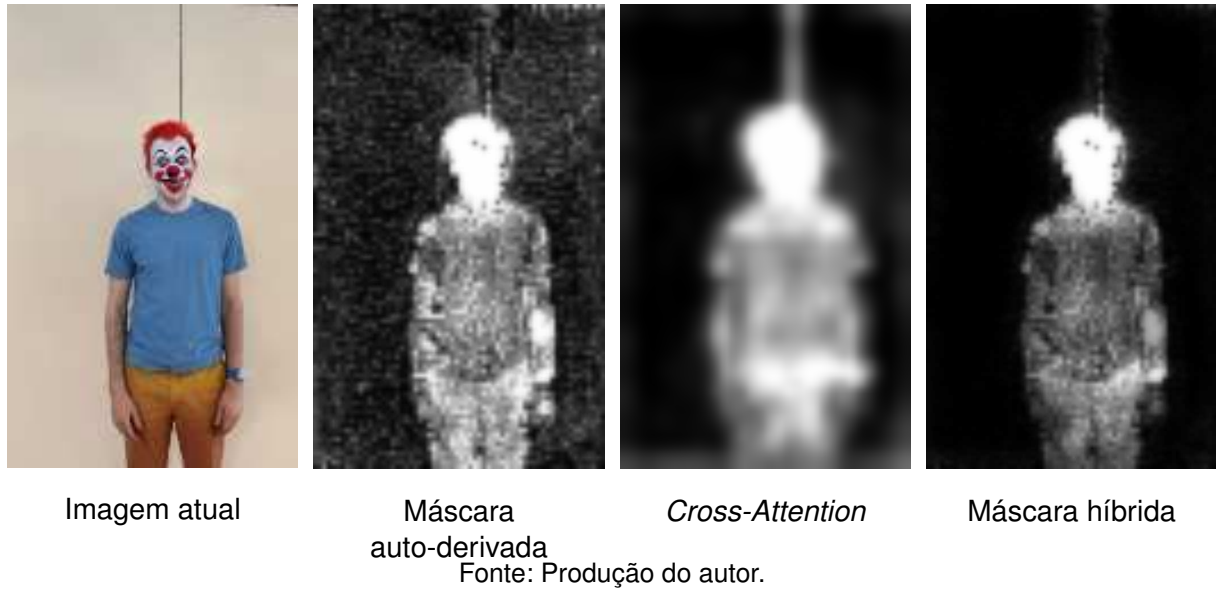


Figura 14 – Exemplo qualitativo de vista renderizada e mapas de localização usados pelo método.



é associado à classe alvo. Os mapas de atenção desses *tokens* são agregados em uma máscara M_{attn} .

A máscara híbrida combina a máscara numérica e a semântica:

$$M_{\text{hybrid}} = M_{\text{self}} \left((1 - w_{\text{CA}}(\hat{t})) + w_{\text{CA}}(\hat{t}) M_{\text{attn}} \right), \quad (20)$$

na qual $w_{\text{CA}}(\hat{t})$ controla a influência da *Cross-Attention* no *timestep* normalizado $\hat{t}$

usado pelo DDS. Essa forma preserva a máscara auto-derivada como sinal principal de onde a edição afeta o *score* e permite que a atenção textual reduza regiões semanticamente menos relevantes.

Em edições criativas, como transformar uma pessoa em stormtrooper, a *Cross-Attention* associada a t_{tgt} pode, inicialmente, concentrar-se nas regiões que já parecem semanticamente próximas do alvo, como o torso, enquanto partes que surgem mais tarde, como o capacete, ainda recebem pouca atenção. Por isso, esta versão implementa uma agenda para o peso da *Cross-Attention* por meio de uma curva reversa ao TAG dentro da faixa ativa de *timesteps*:

$$p(\hat{t}) = \frac{\hat{t}_{\max} - \hat{t}}{\hat{t}_{\max} - \hat{t}_{\min}}, \quad w_{CA}(\hat{t}) = p(\hat{t})^{\kappa e}. \quad (21)$$

Aqui, $\hat{t}$ é o *timestep* normalizado do passo DDS e diminui ao longo da otimização, partindo de $\hat{t}_{\max}$ e chegando a $\hat{t}_{\min}$. O termo κ controla quão tardia é a entrada da *Cross-Attention* ($\kappa=0,75$ na configuração atual). Assim, no início, em *timesteps* mais ruidosos, TAG fica mais forte e a *Cross-Attention* fica fraca; depois, em *timesteps* menos ruidosos, o peso da *Cross-Attention* cresce, permitindo que a máscara semântica refine a localização e reduza regiões de fundo.

4.6 LOCALIZAÇÃO POR MISTURA FONTE-ALVO

Em vez de multiplicar diretamente o gradiente final pela máscara, a abordagem principal mistura as predições de ruído antes de formar o DDS:

$$\epsilon_{tgt}^{loc} = \epsilon_{src} + M \odot (\epsilon_{tgt} - \epsilon_{src}). \quad (22)$$

Fora da máscara, a predição alvo torna-se igual à predição fonte, fazendo com que a diferença DDS tenda a zero. Dentro da máscara, a diferença alvo-fonte é preservada.

4.7 ÂNCORAS DE PRESERVAÇÃO

O DreamCatalyst já possui um termo de preservação baseado em x_0 . Este trabalho reforça esse termo fora da máscara:

$$\psi_{eff} = \psi + w_{out}(1 - M). \quad (23)$$

Assim, regiões com baixa relevância de edição recebem maior força de preservação. Além disso, adiciona-se uma âncora de média latente:

$$g_{final} = g_{DDS} + \lambda \left(\bar{x}_0^{tgt} - \bar{x}_0^{src} \right), \quad (24)$$

com o objetivo de reduzir deslocamentos globais nos canais latentes.

4.8 ADAPTAÇÃO POR FORÇA DE EDIÇÃO

Para adaptar automaticamente a força de preservação a diferentes tipos de *prompt*, calcula-se uma razão de força de edição, aqui denotada por s (reaproveitando o símbolo; distinto da escala de *guidance* do Capítulo 2):

$$s = \frac{\| \epsilon_{tgt}^{raw} - \epsilon_{src}^{raw} \|}{\| \epsilon_{tgt}^{raw} \| + \| \epsilon_{src}^{raw} \|}. \quad (25)$$

Pela desigualdade triangular, $s \in [0, 1]$. Edições mais sutis tendem a ter menor diferença entre alvo e fonte; edições estruturais tendem a ter diferença maior. O método usa $(1 - s)$ para atenuar automaticamente pesos de preservação e STG quando a edição é mais forte.

4.9 MÁSCARA 3D POR VOXEL CACHE

Este módulo é a contribuição da Parte 2: um módulo *opcional* sobre a configuração universal, avaliado separadamente no Capítulo 6. A motivação é simples: se duas câmeras observam o mesmo ponto físico da cena, a decisão de editar ou preservar esse ponto não deveria depender apenas da vista atual. O modo de falha que motivou o módulo é concreto e ocorre tanto no DreamCatalyst base quanto na configuração universal da Parte 1 (sem o *voxel cache*): na cena *clown*, um braço pode ser pintado de vermelho quando observado de um ângulo e permanecer com tom de pele quando observado de outro (Figura 16). O módulo usa profundidade renderizada, raios de câmera e uma grade de *voxels* para transformar um sinal por vista em um sinal 3D consultável pelas demais vistas, de modo que essa decisão passe a ser compartilhada.

Esse módulo deve ser entendido como uma exploração complementar à configuração universal de *guidance*. O objetivo não é afirmar que o DreamCatalyst original sempre produz inconsistência entre vistas, mas tratar uma fragilidade natural de métodos guiados por difusão 2D: cada câmera recebe um sinal próprio, e sinais fortes de TAG/STG podem amplificar pequenas diferenças entre vistas.

O desenho aqui descrito é o resultado de um processo iterativo que faz parte dos resultados do trabalho. A primeira versão agregava diretamente a máscara híbrida 2D e a comparava com a máscara da vista atual; essa formulação expôs um descasamento de escala (o valor consultado é uma *média* multivista comprimida, enquanto a máscara 2D é um indicador nítido) que fazia o termo subtrativo erodir a edição genuína e, após a correção, tornava a consulta redundante. A versão final, denominada ao longo do texto pacote de *força invariante à vista*, altera quatro decisões: *o que* é elevado a 3D (a magnitude bruta de edição, com normalização robusta por quantil), *como* a concordância entre vistas é estimada (variância com ponderação exponencial), *como* o valor é lido de volta (interpolação trilinear ponderada por observação) e *como* ele

é fundido (fusão apenas positiva com *gates* de confiança). A Parte 2 do Capítulo 6 caracteriza o regime de utilidade por cena resultante.

Para cada pixel com profundidade renderizada $D(u, v)$, usa-se a equação paramétrica do raio em coordenadas de mundo:

$$\mathbf{x}_{\text{world}}(u, v) = \mathbf{o}(u, v) + D(u, v)\mathbf{d}(u, v), \quad (26)$$

em que $\mathbf{o}(u, v)$ e $\mathbf{d}(u, v)$ são, respectivamente, a origem e a direção (unitária) do raio da câmera v , obtidas de seus parâmetros intrínsecos e extrínsecos pelo gerador de raios do Nerfstudio, e $D(u, v)$ é a profundidade renderizada esperada *ao longo do raio* (e não a coordenada z), a mesma grandeza definida na renderização volumétrica do Capítulo 2. O ponto é então discretizado em uma célula da grade:

$$\mathbf{q} = \left\lfloor \frac{\mathbf{x}_{\text{world}} - \mathbf{x}_{\min}}{\mathbf{x}_{\max} - \mathbf{x}_{\min}} V \right\rfloor, \quad (27)$$

na qual V é a resolução por eixo da grade e $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ é a caixa 3D usada pelo *voxel cache*. Na implementação atual, essa caixa pode ser estimada a partir dos próprios pontos retroprojetados nas primeiras iterações, com quantis para reduzir a influência dos pontos extremos de profundidade.

O que o *cache* agrega. O sinal elevado a 3D não é a máscara híbrida final, e sim a *magnitude bruta de edição* de cada vista, ou seja, o mapa $R = \|\epsilon_{\text{tgt}}^{\text{raw}} - \epsilon_{\text{src}}^{\text{raw}}\|_2$ já usado pela máscara auto-derivada, antes de *blur*, γ e da modulação semântica. Essa escolha evita que agendas temporais da máscara (como o peso escalonado da *Cross-Attention*) contaminem a estatística multivista. A escala por quadro usa um quantil robusto em vez do máximo:

$$\bar{R}(u, v) = \min \left(1, \frac{R(u, v)}{q_{0,95}(R)} \right), \quad (28)$$

na qual $q_{0,95}(R)$ é o percentil 95 do mapa na vista atual. Normalizar pelo máximo tornava a escala do quadro inteiro refém de um único pixel quente, o que injetava variância espúria entre vistas (duas câmeras concordando sobre a cena, mas divididas por máximos diferentes, pareciam discordar); o quantil elimina essa fonte de falso desacordo.

Como vários pixels de uma mesma vista podem cair no mesmo *voxel*, primeiro calcula-se a média por vista $\bar{M}_v(\mathbf{q})$ dos valores $\bar{R}$. A atualização da grade usa EMA apenas quando essa média passa por um limiar de evidência τ_{upd} :

$$m_{k+1}(\mathbf{q}) = \begin{cases} \beta_{\text{EMA}} m_k(\mathbf{q}) + (1 - \beta_{\text{EMA}}) \bar{M}_v(\mathbf{q}), & \bar{M}_v(\mathbf{q}) \geq \tau_{\text{upd}}, \\ m_k(\mathbf{q}), & \bar{M}_v(\mathbf{q}) < \tau_{\text{upd}} \end{cases}. \quad (29)$$

Esse limiar evita que regiões que ainda não receberam sinal de edição acumulem um valor baixo cedo demais e passem a suprimir edições que surgem apenas no meio

da otimização (quando ativo, ele bloqueia também as estatísticas de confiança, não apenas a média da grade). Trata-se de um mecanismo *disponível*, porém desligado na configuração final ($\tau_{\text{upd}} = 0$): a janela de *warmup* tardia da fusão, descrita adiante, cumpre esse papel de proteger o início do treinamento. *Voxels* observados pela primeira vez copiam diretamente o valor da vista, sem mistura com o valor de *fallback*. O valor de β_{EMA} também pode ser ajustado ao número de câmeras:

$$\beta_{\text{EMA}} = 1 - \frac{1}{cN_{\text{cam}}}, \quad (30)$$

em que N_{cam} é o número de vistas de treinamento e c controla a memória do *voxel cache*. Assim, cenas com mais câmeras usam uma média mais estável.

Além do valor médio $m(q)$, o *voxel cache* mantém estatísticas de confiança por *voxel*: número de vistas únicas que observaram a célula e variância entre as médias observadas. O limiar mínimo de observações é escolhido por fração do número de câmeras, com limites inferior e superior:

$$n_{\min} = \text{clamp}(\lceil rN_{\text{cam}} \rceil, n_{\text{floor}}, n_{\text{cap}}). \quad (31)$$

A confiança usada na fusão é então

$$C(q) = 1[n(q) \geq n_{\min}] \cdot \max\left(0, 1 - \frac{\text{Var}(q)}{\sigma_{\max}^2}\right). \quad (32)$$

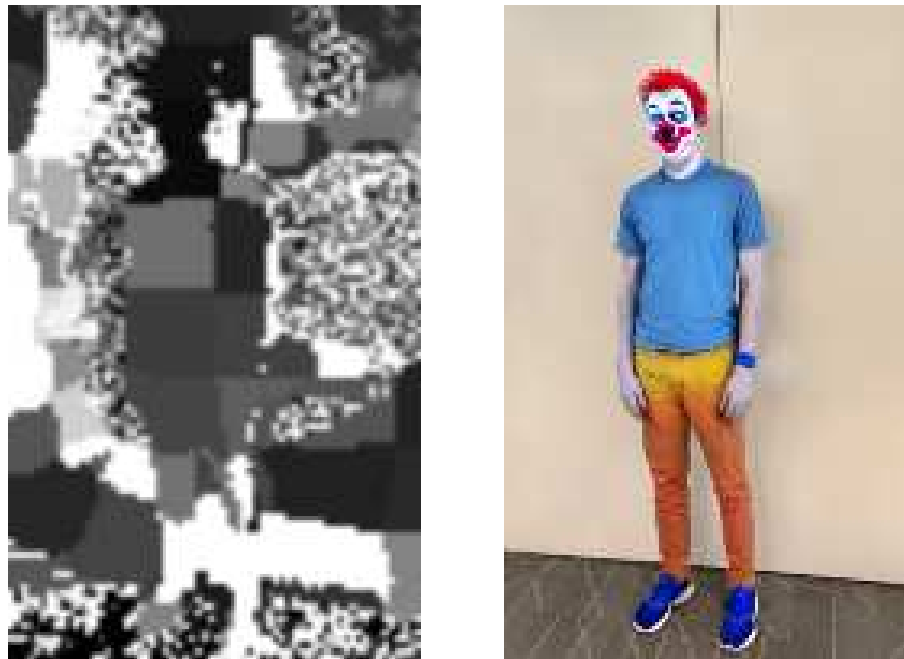
Essa confiança não significa que uma região deva ser editada; significa apenas que o valor do *voxel cache* é confiável naquele ponto 3D. A função indicadora $1[\cdot]$ vale 1 quando a condição entre colchetes é verdadeira e vale 0 caso contrário. Portanto, se um *voxel* foi observado por menos de $n_{\min}$ vistas, sua confiança é zerada, ainda que o valor médio da máscara pareça alto. Quando o número mínimo de vistas é atingido, a segunda parte da equação reduz a confiança de *voxels* com alta variância entre vistas. Por isso, o fundo pode ter alta confiança quando muitas vistas concordam que seu valor de máscara é baixo. Para que vistas repetidas não esvaziem artificialmente a variância, cada câmera contribui com *uma única* amostra por *voxel* (a primeira observação); revisitas atualizam o valor médio da grade, mas não as estatísticas de confiança.

Variância com ponderação exponencial. A estatística cumulativa de Welford pondera igualmente todas as vistas ao longo de *todo* o treinamento. A edição, porém, é não estacionária: um *voxel* observado pela câmera A no início do treinamento (edição imatura) e pela câmera B mais tarde (edição madura) registra a diferença $A \leftrightarrow B$ como “variância entre vistas”, quando ela é, na verdade, deriva *temporal* da própria edição. O sintoma observado é a variância média crescer monotonicamente ao longo da execução, estrangulando o *gate* tardiamente. Por isso, a versão final usa uma variância com ponderação exponencial (Finch, 2009): a cada nova amostra x de um *voxel*, com peso α atribuído ao exemplo mais novo,

$$\delta = x - \mu, \quad \mu \leftarrow \mu + \alpha \delta, \quad \sigma^2 \leftarrow (1 - \alpha) (\sigma^2 + \alpha \delta^2). \quad (33)$$

A memória efetiva é de aproximadamente $1/\alpha$ vistas ($\alpha=0,2$ corresponde às últimas cinco), o que mantém a estatística acompanhando o desacordo *atual* em vez de acumular a evolução histórica da edição. O estimador é enviesado para baixo por um fator $\approx (1 - \alpha)$, de modo que o limiar $\sigma_{\max}^2$ precisa ser reajustado ao habilitá-lo. Na prática, $\sigma_{\max}^2$ é ajustado por cena a partir da leitura do mapa de variância de depuração, posicionando-o entre a variância da região de edição desejada (baixa) e a da região mais propensa à sobre-edição (mais alta). A Figura 15 mostra esse mapa na cena *clown*, ao lado da vista editada correspondente: a cabeça aparece escura e os braços, um pouco mais claros.

Figura 15 – Mapa de variância entre vistas do *voxel cache* (esquerda), normalizado por $\sigma_{\max}^2$, ao lado da vista editada correspondente (direita) na cena *clown*. A cabeça aparece escura (baixa variância entre vistas: edição desejada e coerente), enquanto os braços ficam *um pouco mais claros*: variância mais alta que a da cabeça, sinalizando a região mais propensa à sobre-edição, ainda que nesta execução ela não tenha ocorrido. O contraste é modesto, mas suficiente para posicionar $\sigma_{\max}^2$ no vão entre as duas regiões; é esse o instrumento de ajuste por cena do *gate* de concordância.



Mapa de variância

Vista editada correspondente

Fonte: Produção do autor.

4.9.1 Diversidade angular como fator adicional de confiança

A confiança $C(q)$ definida acima é necessária, mas não suficiente. A variância entre vistas só é informativa quando as vistas observam o *voxel* a partir de *direções angularmente distintas*; vistas próximas e quase paralelas concordam por construção, independentemente do conteúdo real do *voxel*. Empiricamente, na cena *elf* (65 câme-

ras com configuração relativamente agrupada) a fusão exibia variância média $\approx 0,005$ e cobertura de confiança $> 0,5$ acima de 0,98, mas degradava o resultado por erosão suave do alvo: o *cache* confiava em *voxels* observados apenas a partir de um cone estreito de direções. A variância isolada não detecta esse modo de falha, pois raios paralelos concordam mesmo sem triangulação real.

Para distinguir “concordância por triangulação ampla” de “concordância por direções agrupadas”, usa-se uma medida de estatística circular: o comprimento do resultante das direções unitárias de observação (Fisher, 1953). Seja $\hat{d}_v(q)$ a direção unitária média dos raios da vista v que atingem o *voxel* q . Mantém-se um acúmulo $S(q) = \sum_v \hat{d}_v(q)$ atualizado apenas em observações de vistas inéditas (mesmo *gate* da estatística de Welford). O comprimento resultante normalizado vale

$$\ell(q) = \frac{\|S(q)\|}{n(q)} \in [0, 1]. \quad (34)$$

$\ell(q) = 1$ corresponde a raios todos paralelos (sem informação de triangulação) e $\ell(q) \rightarrow 0$ a uma distribuição uniforme na esfera (triangulação máxima). O fator de diversidade angular é, portanto, $A(q) = 1 - \ell(q)$.

Na prática, $1 - \ell(q)$ é uma quantidade dependente do equipamento de captura. Para a cena *clown* (365 câmeras em órbita horizontal sem diversidade vertical), a média de $1 - \ell$ sobre *voxels* multi-vista é $\approx 0,05$; para a cena *elf* (65 câmeras em configuração frontal), $\approx 0,01$. Usar A diretamente faz o *gate* colapsar a contribuição do *cache* em ambas as cenas, embora a configuração da *clown* ofereça muito mais informação de triangulação por *voxel* do que a da *elf*. Aplica-se, então, uma normalização relativa ao cenário:

$$\tilde{A}(q) = \min \left(1, \frac{1 - \ell(q)}{\bar{A}} \right), \quad \bar{A} = \mathbb{E}_{q: n(q) \geq n_{\min}} [1 - \ell(q)]. \quad (35)$$

$\bar{A}$ é a média do fator angular sobre os *voxels* confiáveis da cena, ou seja, aqueles com $n(q) \geq n_{\min}$, a mesma população que o *gate* de confiança já considera estável. Um *voxel* com $\tilde{A}(q) > 1$ tem triangulação superior à média da própria cena (clipado a 1), e $\tilde{A}(q) < 1$ marca *voxels* com observação angularmente mais agrupada que o típico daquela captura. Empiricamente, calcular $\bar{A}$ a cada *query* faz o denominador derivar para baixo ao longo do treinamento, conforme *voxels* de borda alcançam $n_{\min}$ com observações angularmente pobres; isso eventualmente satura $\tilde{A}(q) \approx 1$ em todos os *voxels* e reduz o *gate* a uma operação identidade. Por isso, $\bar{A}$ é congelado de forma *adaptativa à cena*: o módulo acompanha o valor máximo da média sobre a população confiável e, quando esse pico deixa de melhorar por um número fixo de passos (paciência), congela $\bar{A}$ no valor do pico, reutilizando-o como constante pelo restante do treinamento. Esse critério de “paciência sem melhora” captura o pico no instante correto em cada cena (cedo em capturas orbitais densas, mais tarde em capturas frontais), evitando o problema de um passo de congelamento fixo que acerta

uma cena e erra outra. A confiança usada na fusão passa a ser

$$\tilde{C}(q) = C(q) \cdot \tilde{A}(q). \quad (36)$$

Esse uso da média do cenário transforma o *gate* de uma medida absoluta de qualidade de triangulação em uma medida relativa, deixando a unidade comparável entre capturas e impedindo que o *gate* suprima indiscriminadamente o *cache* em cenas cuja geometria de captura naturalmente concentra raios numa região menor da esfera unitária.

O custo de memória do estado portador do *gate* é de $3V^3$ floats além do estado existente (um somatório vetorial por *voxel*). Na configuração de produção ($V = 64$), isso corresponde a aproximadamente 3 MB extras.

4.9.2 Leitura do *cache* e preparação da máscara 3D

Em vistas futuras, a mesma retroprojeção consulta o *voxel cache* e gera M_{3D} . Duas etapas de preparação tornam essa leitura utilizável.

Leitura trilinear ponderada por observação. Em vez de ler apenas o *voxel* mais próximo, o valor é interpolado trilinearmente entre os oito *voxels* vizinhos do ponto 3D, com cada vértice ponderado pelo fato de já ter sido observado. Dessa forma, *voxels* nunca vistos (que carregam apenas o valor de *fallback*) não contaminam o resultado. Essa suavização remove artefatos de discretização em blocos nas fronteiras entre *voxels*, que, de outro modo, aparecem como falso desacordo entre vistas; é a etapa que converte consistência no nível da máscara em consistência efetivamente *renderizada* (Parte 2). As estatísticas de confiança (contagem, variância, diversidade angular e validade) continuam sendo lidas no *voxel* mais próximo, de modo que a semântica desses *gates* não muda; apenas o *gate* de massa, por operar sobre o próprio valor lido, passa a consumir o valor interpolado.

Casamento de escala. O valor consultado é uma média multivista comprimida em torno de valores intermediários, enquanto a máscara 2D é um indicador nítido em $[0, 1]$. Compará-los diretamente foi a origem do descasamento de escala mencionado na introdução desta seção. Antes da fusão, o valor consultado é, portanto, reescalado para $[0, 1]$ sobre a faixa ativa dos *voxels* observados na vista atual:

$$M_{3D} \leftarrow \text{clamp} \left(\frac{M_{3D} - q_{1-p}}{q_p - q_{1-p}}, 0, 1 \right), \quad (37)$$

na qual q_p e q_{1-p} são percentis (com $p=0,95$) calculados sobre os pixels válidos da vista. Com a comparação em unidades equivalentes, a diferença ΔM passa a refletir desacordo real, e não um artefato de unidades.

Gate de massa. Por fim, a confiança é modulada adicionalmente por $C_{\text{mass}} = \min(1, m(q)/\tau_m)$ (na configuração final, $\tau_m = 0,18$), em que $m(q)$ é o valor médio

armazenado. *Voxels* com massa baixa são regiões em que o modelo de difusão não está depositando sinal de edição; permitir a contribuição do *cache* nessas regiões produz artefatos geométricos em extremidades e borra detalhes finos.

4.9.3 Fusão apenas positiva

A fusão final compara as duas máscaras,

$$\Delta M = M_{3D} - M_{\text{hybrid}}, \quad (38)$$

e, na configuração final, usa *apenas* o ramo positivo:

$$M_{\text{final}} = M_{\text{hybrid}} + \alpha(t) \tilde{C} C_{\text{mass}} G \max(\Delta M, 0). \quad (39)$$

Aqui $\alpha(t)$ é o peso do *cache* (reaproveitando o símbolo α ; distinto do fator de ponderação da variância exponencial), que cresce linearmente de zero ao valor máximo durante a janela de *warmup*; $\tilde{C}$ é a confiança combinada (contagem $\times$ variância $\times$ diversidade angular); C_{mass} é o *gate* de massa definido acima; e G é um *gate* semântico para contribuições positivas, $G = \text{clamp}(M_{\text{attn}} + (M_{\text{self}} - M_{\text{attn}})^+, 0, 1)$, equivalente a $\max(M_{\text{attn}}, M_{\text{self}})$: abrir novas regiões de edição exige confirmação semântica ou numérica da própria vista. Portanto, o *cache* apenas *soma* força de edição onde as vistas concordam, nunca a subtrai. Um ramo subtrativo foi avaliado em versões anteriores e descartado, pois é multiplicado pela mesma confiança $\tilde{C}$ (aproximadamente nula onde o *gate* desconfia) e, por construção, abstém-se exatamente onde seria necessário.

Para fins de depuração, a implementação registra em WandB mapas como M_{self} , M_{attn} , M_{final} , M_{3D} , o mapa de variância normalizado por σ_{max}^2 (o instrumento de ajuste por cena do *gate* de concordância) e a confiança do *voxel cache*.

Na Figura 10, o termo de correção da Equação (39) aparece de forma condensada como $\text{correction}(M_{3D} - M_{\text{hybrid}})$: a correção ponderada por confiança, que *eleva* a máscara onde o *voxel cache* confia mais que a máscara 2D híbrida. Quando o *voxel cache* está desativado (o caso da configuração universal da Parte 1), não há esse termo e $M_{\text{final}} = M_{\text{hybrid}}$, de modo que a localização por mistura fonte-alvo e a âncora de fundo usam diretamente a máscara 2D híbrida.

A Figura 16 ilustra a motivação qualitativa desse módulo: uma edição vista a vista pode produzir inconsistência entre ângulos, enquanto a agregação 3D busca compartilhar evidências entre vistas. A figura serve como motivação visual do módulo; os resultados quantitativos correspondentes são apresentados na Parte 2 do Capítulo 6.

Definidos os módulos da solução proposta (direcionamento, localização, preservação e a memória 3D opcional), o Capítulo 5 descreve o protocolo de avaliação: as cenas e instruções, as métricas reportadas e os critérios adotados para medir a contribuição de cada componente.

Figura 16 – Motivação visual para o *voxel cache* 3D: sem agregação 3D, a mesma região física (o braço) pode ser editada em uma vista e preservada em outra; a agregação em *voxels* compartilha a decisão entre as câmeras. A Parte 2 do Capítulo 6 mostra que o *cache* de fato restaura essa coerência entre vistas e caracteriza o regime de utilidade resultante.



Edição por máscaras 2D por vista

Com *voxel cache* 3D (ver Parte 2)

Fonte: Produção do autor.

5 METODOLOGIA DE AVALIAÇÃO

Este capítulo descreve a metodologia de avaliação adotada. São detalhadas as comparações realizadas, as cenas e instruções empregadas, as métricas reportadas e o tipo de figura qualitativa que acompanha os resultados numéricos. Os resultados obtidos com esse desenho experimental são apresentados e discutidos no Capítulo 6.

5.1 PROTOCOLO EXPERIMENTAL

Cada experimento parte de uma cena 3D reconstruída previamente e aplica uma instrução textual de edição usando o mesmo fluxo geral descrito no Capítulo 4. Para que a comparação seja justa, reconstrução, edição e avaliação devem usar o mesmo fator de escala das imagens sempre que possível. Quando esse fator muda, a execução é tratada como uma configuração distinta, pois a qualidade da reconstrução e a resolução das vistas renderizadas podem afetar diretamente as métricas.

A avaliação é organizada em duas partes. A primeira avalia os módulos de *guidance*, localização e preservação como uma configuração universal, isto é, uma única configuração fixa aplicada a múltiplas cenas. A segunda investiga se o *voxel cache*, uma memória 3D da magnitude de edição observada por vista, pode promover consistência entre vistas, em estudo separado da conclusão principal sobre a configuração universal. Essa separação evita que dois objetivos distintos se confundam: de um lado, melhorar a qualidade geral da edição; de outro, investigar se essa memória 3D reduz inconsistências entre câmeras.

O método permanece modular. Com os componentes novos desativados, o fluxo deve reproduzir o DreamCatalyst base; ativando subconjuntos dos módulos, é possível medir suas contribuições individuais por ablação.

5.2 CONFIGURAÇÕES EXPERIMENTAIS

Dentro dessa estrutura, as configurações comparadas organizam-se em três níveis:

- **DreamCatalyst base** (*baseline*): sem os módulos adicionais;
- **Ablações** (*Ablations*): ativações parciais dos módulos, como TAG, STG, máscara auto-derivada, *Cross-Attention* e âncora de fundo (o *voxel cache* 3D é avaliado à parte, na Parte 2);
- **Configuração universal**: uma única configuração fixa testada em múltiplas cenas e *prompts*.

Além disso, algumas configurações ajustadas por cena são empregadas para investigar o limite superior de qualidade visual em cada tipo de edição. Esses ajustes não são tratados como resultado principal, mas como análise do comportamento do método.

5.3 CENAS E *PROMPTS*

A campanha experimental cobre cenas com diferentes níveis de dificuldade:

- edições de identidade localizadas (inclusive identidade com fantasia), nas quais a região relevante é localizada e a preservação é crítica;
- edições estruturais de corpo inteiro, nas quais o *prompt* exige mudança estrutural maior e há maior risco de inconsistência entre vistas, motivando a investigação do *voxel cache* 3D;
- adições estruturais localizadas, como acrescentar um capacete a uma face;
- edições objeto-cêntricas com fundo visível, para avaliar a preservação de geometria e textura do fundo.

Os nomes das cenas, as instruções e os pontos de controle (*checkpoints*) utilizados são apresentados no Capítulo 6 (Tabela 1), de modo a permitir a reprodutibilidade.

5.4 MÉTRICAS

As métricas quantitativas reportadas são:

- **CLIP Direction**: mede se a direção da mudança visual acompanha a direção da mudança textual; é a métrica principal de *editabilidade*, isto é, da eficácia da edição em seguir o comando textual;
- **CLIP Image Sim** (*CLIP-I*): mede proximidade entre a imagem editada e a imagem fonte;
- **SSIM**: mede similaridade estrutural;
- **LPIPS**: mede distância perceptual aprendida;
- **MV Pairwise Cos Sim** (*MV Cos*): estima coerência entre vistas renderizadas da cena editada;
- **EditMaskVariance_3D**: métrica de consistência multivista proposta neste trabalho. Para cada vista, mede-se quanto cada pixel mudou com a edição (a magnitude da diferença entre a vista editada e a vista fonte); essas magnitudes são

retroprojetadas para uma grade 3D pela profundidade renderizada e, em cada *voxel*, calcula-se a variância entre as vistas que o observam. Como o sinal vem das imagens renderizadas, e *não* da máscara usada no treinamento, é um *proxy post-hoc*: mede a consistência do resultado, não a do mecanismo interno. Trata-se, porém, de uma métrica confundida: a sobre-edição (que torna a edição mais uniforme) reduz sua variância sem melhorar a consistência genuína, de modo que “menor = melhor” não se sustenta isoladamente. Por isso, a métrica é interpretada com cautela e em conjunto com a *MV Pairwise Cos Sim*, e não como critério único.

Essas métricas são interpretadas em conjunto com inspeção visual. Em edição 3D, um resultado pode apresentar métrica favorável e, ainda assim, exibir falhas semanticamente importantes, como assimetria entre membros, alterações no fundo ou perda de detalhes em regiões pequenas.

5.5 CRITÉRIOS DE INTERPRETAÇÃO

A análise dos resultados é guiada pelas seguintes perguntas:

1. O método proposto preserva melhor regiões fora da edição?
2. A localização por máscara reduz alterações indesejadas?
3. O *guidance* adaptativo aumenta a força da edição sem degradar estrutura?
4. Em que regime uma memória 3D de evidências de edição (*voxel cache*) promove consistência entre vistas? Mais especificamente: em quais tipos de cena ela ajuda, quais *gates* de concordância são necessários para que o *cache* não amplifique sinais coerentes mas indesejados, e o que isso revela sobre o papel da máscara de localização na consistência multivista?
5. A configuração universal se mantém competitiva em cenas diferentes?

Essas perguntas estruturam a discussão apresentada no Capítulo 6.

5.6 LIMITAÇÕES DA AVALIAÇÃO

O desenho experimental adotado é adequado para um trabalho de conclusão de curso, mas três limitações devem ser explicitadas e consideradas na leitura das conclusões:

- **Número de execuções.** Cada configuração é avaliada com $n = 3$ sementes, o que torna médias e desvios mais confiáveis do que uma execução única, mas

oferece poucos graus de liberdade para a estimativa de variância. Por isso, apenas diferenças que excedem o piso de ruído de execução para execução são tratadas como reais; diferenças marginais são reportadas como inconclusivas (Capítulo 6).

- **Número e diversidade de cenas.** A campanha cobre oito edições sobre um conjunto reduzido de cenas, escolhidas para representar eixos distintos de dificuldade. Esse conjunto é suficiente para caracterizar tendências, mas pequeno para afirmar generalização ampla a outras cenas, objetos e categorias de edição.
- **Dependência dos *prompts*.** Como a edição é condicionada por texto, as conclusões valem para as instruções de fonte e alvo efetivamente avaliadas (Tabela 5). Reformulações de *prompt* podem deslocar o equilíbrio entre editabilidade e preservação, e a sensibilidade a essas reformulações não foi medida de forma sistemática.

6 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados da campanha experimental descrita no Capítulo 5. A avaliação segue a separação em duas partes já estabelecida: a Parte 1, que concentra o resultado central do trabalho, mede a configuração universal de *guidance*, localização e preservação contra o DreamCatalyst base; a Parte 2 avalia o *voxel cache* 3D como módulo opcional, uma investigação complementar, e o caracteriza por um *regime de utilidade por cena*. Nas métricas agregadas, o módulo é neutro; seu mecanismo de consistência multivista evidencia-se apenas em isolamento controlado (cenas de detalhe fino), ao passo que cenas de recoloração ampla exigem *gates* de concordância entre vistas. A Parte 2 reúne ainda os achados sobre os mecanismos do método que explicam esse regime. Os números são acompanhados de inspeção visual, pois em edição 3D uma métrica favorável pode coexistir com falhas semanticamente relevantes.

6.1 CONFIGURAÇÃO EXPERIMENTAL

Para isolar o efeito da configuração de edição, todas as execuções de uma mesma cena partem do *mesmo* ponto de controle (*checkpoint*) de reconstrução *nerf-facto*; apenas a configuração de edição varia entre o DreamCatalyst base e o método proposto. Reconstrução, edição e avaliação usam o mesmo fator de escala das imagens. O *guidance scale* é mantido fixo em 7,5 em todas as execuções, reproduzindo o valor do DreamCatalyst base para uma comparação justa. Todos os experimentos (reconstrução, edição e avaliação) foram conduzidos em uma GPU NVIDIA H100.

Cada configuração é executada $n = 3$ vezes por cena. Como o processo de destilação é estocástico (sequência de ruído, seleção de vistas e não determinismo de *kernels* em GPU), execuções com a mesma configuração não são idênticas. Mede-se empiricamente uma variação de execução para execução da ordem de 0,005 a 0,01 em *CLIP Direction* neste *pipeline*; por esse motivo, os resultados são resumidos pela média sobre as três execuções, com a dispersão entre execuções usada para avaliar a significância das diferenças, e não por uma única execução. Esse desenho evita que uma única execução favorável (ou desfavorável) determine a conclusão.

A Tabela 1 lista as oito edições avaliadas nesta versão. Os pontos de controle de reconstrução e os fatores de escala usados por cena são documentados no Apêndice A para permitir reprodutibilidade. As cenas cobrem os quatro eixos de dificuldade pretendidos: edições de identidade localizadas (*face* e *yuseung*), edições estruturais de corpo inteiro (*person*), adições estruturais localizadas (*fangzhou*) e edições objeto-cêntricas (*bear*).

Tabela 1 – Cenas e instruções de edição avaliadas, organizadas pelos eixos de dificuldade definidos no Capítulo 5. A coluna *Sementes* indica o número de execuções por configuração usado em cada cena.

Cena	Instrução de edição	Categoria	Sementes
bear	<i>Turn the bear statue into a panda</i>	Estrutural, objeto-cêntrica	3
face	<i>Turn him into a Tolkien elf</i>	Identidade, localizada, preservação crítica	3
face	<i>Turn him into an Einstein</i>	Identidade, localizada	3
person	<i>Turn him into a clown</i>	Corpo inteiro, trajetória não monotônica	3
person	<i>Turn him into a stormtrooper</i>	Corpo inteiro, estrutural (crescimento de capacete)	3
fangzhou	<i>Put a knight's helmet on him</i>	Adição estrutural localizada	3
yuseung	<i>Turn him into a Batman</i>	Identidade com fantasia	3
yuseung	<i>Turn him into a Joker</i>	Identidade com fantasia	3

Fonte: Produção do autor.

6.1.1 Configuração universal proposta

Todos os resultados da coluna “Proposto” usam uma *única* configuração fixa (Tabela 2), detalhada no Capítulo 4; listam-se apenas os módulos contribuídos por este trabalho, mantendo-se os parâmetros herdados do DreamCatalyst base em seus valores padrão. Os mecanismos de direcionamento respondem pela força de edição: o TAG adaptativo e assimétrico ($\eta = 1,2$) reforça a direção do *score* sem inflar sua norma, e o STG (escala 3,5) usa um escalonamento em *bump* (nulo no início, máximo por volta da metade da otimização, com pico = 0,5, e decaindo até 0,8), ajudando na formação de estrutura na fase intermediária sem congelar estados parciais dependentes da vista no final. STG e a âncora de fundo são modulados pela força de edição ($1 - s$), que reduz automaticamente a amplificação à medida que a edição converge. A localização combina a máscara auto-derivada com a máscara semântica de *Cross-Attention* (peso escalonado, $\kappa = 0,75$) e aplica a edição por *source-blended localization*; a preservação usa âncoras fora da máscara e de média latente.

Quando o *voxel cache* é habilitado (Parte 2), a escala de STG é reduzida de 3,5 para 3,0. Trata-se de um *rebalanceamento de projeto*: o *cache* adiciona um suporte de localização condicionado por concordância entre vistas, de modo que o amplificador de *guidance* é atenuado para evitar a dupla amplificação do mesmo sinal de edição. No agregado, o efeito dessa redução sobre as métricas fica dentro do piso de ruído de execução para execução (Parte 2, Tabela 4); por cena, há um compromisso, analisado na Seção 6.4.1: a menor amplificação contém a sobre-edição em cenas de recoloração ampla (*clown*), mas subedita cenas que dependem de completar estrutura fina (*bear*).

Tabela 2 – Configuração universal proposta (módulos contribuídos por este trabalho; *voxel cache* desligado). Parâmetros herdados do DreamCatalyst base não são listados.

Módulo	Configuração
TAG	$\eta = 1,2$; adaptativo (varia com o <i>timestep</i>) e assimétrico (aplicado ao ramo alvo)
STG	escala 3,5 na Parte 1, reduzida para 3,0 na configuração com <i>voxel cache</i> (Parte 2); escalonamento <i>bump</i> (início 0, pico 0,5, fim 0,8); <i>skip layer</i> 2; composição paralela com TAG; modulado por força de edição
Máscara auto-derivada	<i>blur</i> 0,5; $\gamma = 1,2$; EMA automático por número de câmeras
Máscara de <i>Cross-Attention</i>	escalonamento de potência ($\kappa = 0,75$); camadas [1, 2]; $\gamma = 1,2$
Localização	<i>source-blended localization</i> (porta dura fora da máscara)
Preservação	âncora fora da máscara 0,15 (adaptativa por força de edição); âncora de média latente 0,005

Fonte: Produção do autor.

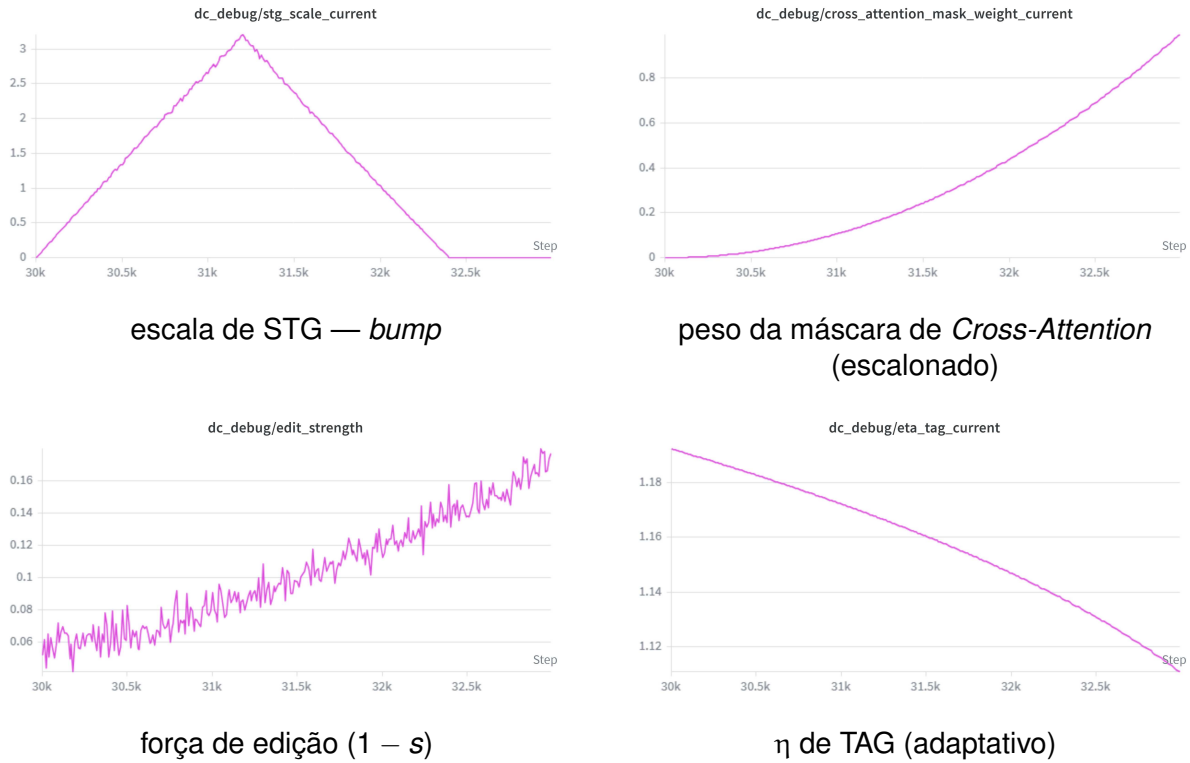
A Figura 17 mostra o comportamento desses mecanismos ao longo do treinamento, exportado do WandB para uma execução representativa da configuração proposta (sem *voxel cache*).

6.2 PARTE 1 — CONFIGURAÇÃO UNIVERSAL VS DREAMCATALYST

A Tabela 3 compara o DreamCatalyst base (DC; todos os módulos adicionais desligados) com a configuração universal proposta. Para cada cena, reporta-se a média sobre $n = 3$ execuções, em cinco métricas. Setas indicam o sentido de melhora ($\uparrow$ maior é melhor; $\downarrow$ menor é melhor). Cada célula mostra a transição DC $\rightarrow$ Proposto; o valor em destaque (verde) indica o vencedor por métrica e cena. Conforme os critérios de significância (parágrafo abaixo), apenas diferenças que excedem a dispersão combinada das três execuções são destacadas como vitória; células cuja diferença fica dentro do piso de ruído permanecem sem destaque (empate efetivo).

A Figura 18 acompanha a tabela com a comparação visual das oito edições, contrastando a renderização fonte, o DreamCatalyst base e a configuração proposta.

Figura 17 – Curvas de treinamento (WandB) da configuração universal proposta, sem *voxel cache*: o escalonamento *bump* de STG, o peso escalonado da máscara de *Cross-Attention*, o envelope de força de edição ($1 - s$) e o η adaptativo de TAG. A chave WandB de cada curva está indicada no painel correspondente.



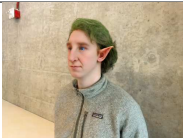
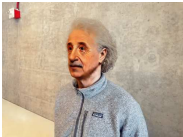
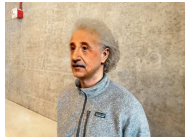
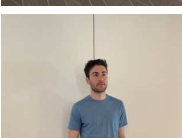
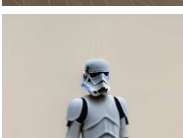
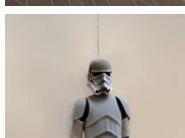
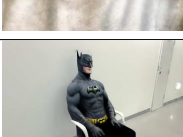
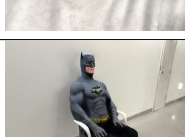
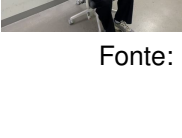
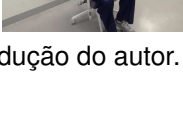
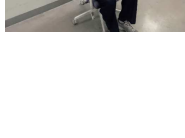
Fonte: Produção do autor.

Tabela 3 – Resultados quantitativos da Parte 1 (configuração universal) na forma DC $\rightarrow$ Proposto. Cada célula mostra a *média* sobre $n = 3$ execuções por cena, com o mesmo *checkpoint* de reconstrução por cena e uma única configuração fixa (sem ajuste por cena). Destaque em **verde** = vencedor quando a diferença excede a dispersão combinada das execuções; células sem destaque são empates dentro do piso de ruído. O desvio padrão por cena é pequeno ($\leq 0,013$, exceto *fangzhou*) e fundamenta o critério de significância adotado; os valores completos de média $\pm$ desvio constam do Apêndice C.

Cena	$CLIP_{dir} \uparrow$ (DC $\rightarrow$ Prop.)	$CLIP-I \uparrow$ (DC $\rightarrow$ Prop.)	$SSIM \uparrow$ (DC $\rightarrow$ Prop.)	$LPIPS \downarrow$ (DC $\rightarrow$ Prop.)	$MV Cos \uparrow$ (DC $\rightarrow$ Prop.)
bear	0,180 $\rightarrow$ 0,174	0,745 $\rightarrow$ 0,788	0,693 $\rightarrow$ 0,768	0,283 $\rightarrow$ 0,234	0,907 $\rightarrow$ 0,896
face (elf)	0,123 $\rightarrow$ 0,128	0,592 $\rightarrow$ 0,601	0,820 $\rightarrow$ 0,834	0,256 $\rightarrow$ 0,231	0,920 $\rightarrow$ 0,928
face (Einstein)	0,081 $\rightarrow$ 0,078	0,557 $\rightarrow$ 0,559	0,794 $\rightarrow$ 0,806	0,255 $\rightarrow$ 0,258	0,922 $\rightarrow$ 0,921
person (clown)	0,138 $\rightarrow$ 0,143	0,773 $\rightarrow$ 0,772	0,924 $\rightarrow$ 0,935	0,227 $\rightarrow$ 0,198	0,911 $\rightarrow$ 0,910
person (stormtrooper)	0,155 $\rightarrow$ 0,171	0,650 $\rightarrow$ 0,682	0,795 $\rightarrow$ 0,839	0,331 $\rightarrow$ 0,295	0,926 $\rightarrow$ 0,901
fangzhou (helmet)	0,206 $\rightarrow$ 0,182	0,727 $\rightarrow$ 0,764	0,796 $\rightarrow$ 0,834	0,311 $\rightarrow$ 0,282	0,951 $\rightarrow$ 0,943
yuseung (Batman)	0,156 $\rightarrow$ 0,165	0,627 $\rightarrow$ 0,653	0,869 $\rightarrow$ 0,889	0,273 $\rightarrow$ 0,251	0,930 $\rightarrow$ 0,935
yuseung (Joker)	0,170 $\rightarrow$ 0,172	0,643 $\rightarrow$ 0,655	0,908 $\rightarrow$ 0,923	0,241 $\rightarrow$ 0,225	0,932 $\rightarrow$ 0,938
Média	0,151 $\rightarrow$ 0,152	0,664 $\rightarrow$ 0,684	0,825 $\rightarrow$ 0,854	0,272 $\rightarrow$ 0,247	0,925 $\rightarrow$ 0,922

Fonte: Produção do autor.

Figura 18 – Resultados qualitativos da Parte 1 (voxel/ cache desligado): renderização fonte, DreamCatalyst base e a configuração proposta. A proposta mantém editabilidade semelhante ao método base, com melhor preservação de identidade, textura e regiões fora da edição (complementa a Tabela 3).

Edição	Fonte	DreamCatalyst base	Proposto universal
Tolkien elf			
			
Einstein			
			
Clown			
			
Stormtrooper			
Panda			
Helmet armor			
			
Batman			
Joker			

Fonte: Produção do autor.

Resultado principal. O padrão é consistente em todo o conjunto: com a *editabilidade praticamente empatada* na média (*CLIP Direction* 0,151 $\rightarrow$ 0,152), a configuração proposta entrega ganhos claros e sistemáticos de *preservação*: SSIM 0,825 $\rightarrow$ 0,854, LPIPS 0,272 $\rightarrow$ 0,247 e *CLIP Image Sim* 0,664 $\rightarrow$ 0,684 na média. A vitória da proposta em SSIM ocorre nas oito edições, e em LPIPS em sete das oito; a *MV Cos* fica essencialmente empatada (0,925 $\rightarrow$ 0,922). Em *CLIP Direction* o resultado é misto e dentro do piso de ruído na maioria das cenas: a proposta vence *stormtrooper*, *elf* e *Batman*, empata em *clown*, *Einstein* e *Joker*, e fica atrás em *bear* e *fangzhou*. Nessas duas, porém, a pequena perda de editabilidade vem acompanhada de um ganho grande de preservação (*bear*: SSIM 0,693 $\rightarrow$ 0,768, LPIPS 0,283 $\rightarrow$ 0,234), o que caracteriza exatamente o compromisso editabilidade–preservação central ao DreamCatalyst: a proposta edita com força comparável, mas preserva identidade e fundo de forma marcadamente melhor.

Leitura conjunta das métricas. A *CLIP Direction* é a métrica principal de editabilidade, mas não deve ser lida isoladamente. Como o objetivo do DreamCatalyst combina editabilidade e preservação de identidade, uma edição mais tímida pode parecer visualmente mais limpa e ainda assim obter *CLIP Direction* menor. Por isso o vetor completo de métricas é reportado: ganhos de preservação (*CLIP Image Sim*, SSIM, LPIPS) com editabilidade equivalente caracterizam a contribuição de localização proposta.

Quando a *CLIP Direction* engana. O caso *fangzhou* ilustra um limite mais forte dessa métrica. Ali o DreamCatalyst base obtém a maior *CLIP Direction* de toda a Tabela 3 (0,206 contra 0,182 da proposta), mas a inspeção visual (Figura 18) mostra que esse valor alto premia justamente o comportamento que se deseja evitar: o método base suaviza fortemente o rosto, perdendo a identidade do sujeito, converte o suéter em uma cota de malha metálica pronunciada e borra a grade de azulejos do fundo, espalhando a transformação “de cavaleiro” para muito além do capacete. A configuração proposta, com *CLIP Direction* menor, preserva bem melhor os traços faciais do sujeito (a textura da pele e marcas específicas, como as pintas na bochecha e no queixo) e adiciona um capacete mais detalhado; o vestuário ainda é empurrado em direção a uma aparência de armadura, porém de forma muito mais fraca e suave que no método base, e o fundo, embora não totalmente preservado nesta cena, permanece mais estável. Por medir o alinhamento textual de forma global, a *CLIP Direction* recompensa a edição mais agressiva e difusa, e não a que preserva melhor a identidade e contém a edição na região pretendida. Esse é um motivo concreto para não tratar a *CLIP Direction* como critério isolado e para acompanhá-la sempre das métricas de preservação e da inspeção visual.

Significância com $n = 3$. Com apenas três execuções por célula, o desvio padrão é estimado com poucos graus de liberdade. Uma melhora é tratada como esta-

belecida apenas quando a diferença entre as médias excede a dispersão combinada (isto é, quando as barras de erro praticamente não se sobrepõem); caso contrário, o resultado é reportado como inconclusivo. Diferenças dentro do piso de ruído (Seção 6.1) não são interpretadas como vantagem do método.

A Figura 19 examina a coerência entre vistas, mostrando a mesma edição renderizada de mais de um ângulo, e não apenas a qualidade de uma imagem isolada.

Figura 19 – Comparação qualitativa entre imagem fonte, DreamCatalyst base e método proposto, em mais de uma vista para evidenciar coerência entre vistas e não apenas a qualidade de uma imagem isolada.

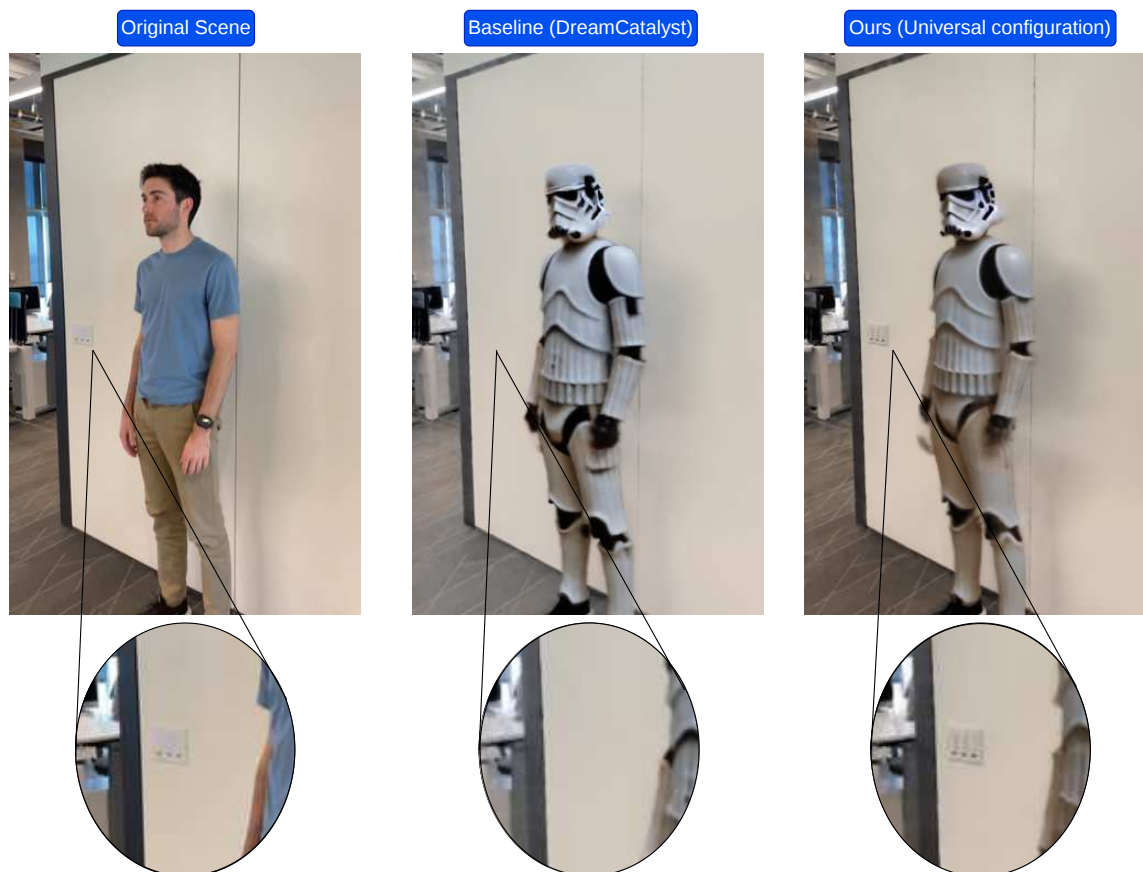


6.2.1 Preservação de regiões fora da edição

Além das métricas de preservação, a diferença aparece visualmente em regiões pequenas do fundo. A Figura 20 mostra um exemplo no qual o DreamCatalyst base remove o interruptor na parede, enquanto a configuração proposta mantém esse detalhe mesmo após a edição semântica do personagem.

Outro exemplo, agora em escala de cena, é a Figura 21, da edição que transforma a estátua de urso em panda. Além de preservar objetos individuais, a configuração proposta mantém o caráter *global* do fundo: a textura da vegetação, a iluminação e as cores permanecem próximas às da reconstrução fonte. O DreamCatalyst base tende a suavizar e deslocar essas regiões (vegetação esmaecida e tonalidade global alterada), produzindo uma aparência menos natural. As três versões são apresentadas lado a lado para permitir a comparação direta da preservação global do fundo. Esse comportamento qualitativo é coerente com a Tabela 3: é em *bear* que a proposta obtém seu maior ganho de preservação (SSIM 0,693 $\rightarrow$ 0,768; LPIPS 0,283 $\rightarrow$ 0,234), ao custo da pequena perda de editabilidade já discutida.

Figura 20 – Exemplo qualitativo de preservação de fundo. Na cena original, há um interruptor visível na parede. O DreamCatalyst base remove ou degrada esse detalhe durante a edição, enquanto a configuração universal proposta preserva melhor a região fora do objeto editado.



Fonte: Produção do autor.

Figura 21 – Comparação qualitativa de preservação global do fundo na cena *bear* (estátua de urso → panda). Da esquerda para a direita: reconstrução fonte, DreamCatalyst base e configuração universal proposta. A proposta mantém textura, iluminação e cores do fundo próximas às da cena fonte, enquanto o método base suaviza a vegetação e desloca a tonalidade global.



Reconstrução fonte

DreamCatalyst base

Configuração universal proposta

Fonte: Produção do autor.

6.3 PARTE 2 — VOXEL CACHE 3D

A Parte 2 avalia o *voxel cache* como módulo opcional sobre a configuração universal. A comparação correta é contra a configuração *standard*, isto é, a mesma configuração 2D com o *cache desligado*, e não contra o DreamCatalyst base. O desfecho não é um ganho uniforme nem um fracasso uniforme, e sim um **regime de utilidade por cena**: nas métricas agregadas o *cache* é *neutro* (todas as seis ficam no piso de ruído, Tabela 4), e seu mecanismo de consistência multivista só se evidencia em isolamento controlado: na cena de detalhe fino (*elf*), a leitura trilinear melhora a consistência renderizada por uma margem pequena, da ordem do piso de ruído; em cenas de recoloração ampla (*clown*), o *cache* amplifica também sinais consistentes e *indesejados*, e o seu uso seguro depende dos *gates* de concordância, cuja análise produziu os achados sobre os mecanismos discutidos nesta seção. Em ambos os regimes, porém, o módulo cumpre a função para a qual foi concebido: compartilhar a decisão de edição entre as câmeras. As falhas residuais tornam-se coerentes entre vistas, deslocando a estocasticidade do ângulo de observação para a execução. A contribuição da Parte 2 é, portanto, a infraestrutura (o *cache*, a métrica *EditMaskVariance_3D* e os painéis de diagnóstico) somada a essa caracterização honesta de regime.

6.3.1 Trajetória de desenvolvimento: da formulação ingênua ao pacote final

A primeira formulação agregava a própria máscara híbrida 2D e a comparava diretamente com a máscara da vista atual. Três achados, todos negativos, motivaram o redesenho e fazem parte dos resultados.

(A) Teste A/B de variância entre vistas nulo. Comparando o *cache* ligado contra um controle passivo (*apenas-medição*, que acumula as mesmas estatísticas sem influenciar o treinamento), a formulação inicial não reduzia a variância *entre vistas* da máscara de relevância, justamente a quantidade que o módulo deveria homogeneizar.

(B) Descasamento de escala e redundância. A máscara consultada é uma *média* multivista comprimida, enquanto a máscara 2D é um indicador nítido em $[0, 1]$; diferenciá-las fazia o termo subtrativo *erodir* a edição genuína. A normalização de escala (Capítulo 4) corrige a aplicação, mas a consulta então apenas reproduzia a máscara 2D.

(C) Não estacionariedade. A magnitude de edição destilada deriva ao longo do treinamento, de modo que a variância cumulativa *cross-view* cresce monotonicamente e estrangula o *gate* de confiança tardiamente.

O pacote final de *força invariante à vista* (Capítulo 4) responde a cada achado: eleva a magnitude bruta de edição com normalização robusta por quantil (remove variância espúria de escala), estima a concordância com variância de ponderação exponencial (remove a deriva temporal da estatística), lê o valor de volta por interpola-

ção trilinear ponderada por observação (remove o falso desacordo de discretização) e funde apenas o ramo positivo, condicionado por confiança.

6.3.2 Resultados quantitativos por cena

A Tabela 4 reúne os resultados da Parte 2 nas oito edições, comparando a configuração 2D da Parte 1 (sem *voxel cache*) com a configuração que habilita o *cache* (Parte 2), ambas com $n = 3$ e o mesmo *checkpoint*. Na média, todas as seis métricas avaliadas ficam dentro do piso de ruído de execução para execução (Seção 6.1): *CLIP Direction* 0,152 $\rightarrow$ 0,153, *CLIP-I* 0,684 $\rightarrow$ 0,685, SSIM 0,854 $\rightarrow$ 0,854, LPIPS 0,247 $\rightarrow$ 0,244, *MV Cos* 0,922 $\rightarrow$ 0,919 e *EditMaskVariance_3D* 0,0166 $\rightarrow$ 0,0164. No nível das métricas, portanto, o *cache* é *neutro* (não há ganho nem perda uniforme), coerente com seu papel de módulo opcional.

Embora o agregado seja neutro, há um padrão por cena na editabilidade: em *CLIP Direction*, a configuração com *cache* é nominalmente melhor em cinco das oito edições (*elf*, *Einstein*, *clown*, *stormtrooper* e *fangzhou*); a média só não sobe porque duas cenas a puxam para baixo: sobretudo a *bear* (0,174 $\rightarrow$ 0,166) e, em menor grau, a edição *Batman*. A queda da *bear* é o efeito esperado da redução de STG, que subedita a recoloração estrutural das patas do panda (Seção 6.4); a LPIPS segue o mesmo sentido, melhorando na maioria das cenas. As demais métricas (*CLIP-I*, *MV Cos* e *EMV_{3D}*) ficam repartidas entre as cenas, sem direção dominante. É importante notar que se trata de uma comparação *entre configurações*: a coluna *voxel* carrega também o rebalanceamento de STG para 3,0 (Seção 6.1.1), de modo que as diferenças misturam o efeito do *cache* e o da escala de STG, e não isolam o *cache*.

O ponto central, porém, não está nas métricas, e sim no problema que o módulo foi concebido para resolver: a *inconsistência entre vistas*. As métricas de imagem não capturam essa mudança: a própria *MV Cos* fica marginalmente menor (dentro do piso de ruído), pois mede a similaridade entre vistas já renderizadas, e não a coerência da *decisão* de edição. Qualitativamente, contudo, com o *cache* as falhas deixam de variar com o ângulo da câmera: a mesma região é editada em todas as vistas ou em nenhuma (Seção 6.3.4). Como essa coerência entre vistas é uma propriedade desejável em edição 3D, e a editabilidade permanece no piso de ruído na maioria das cenas, o uso do *cache* é em geral preferível à sua ausência nas edições para as quais foi projetado, tendo a *bear* (edição estrutural penalizada pela redução de STG) como principal exceção.

6.3.3 Cena de detalhe fino (*elf*): consistência renderizada

O isolamento controlado do efeito do *cache* aparece em uma ablação na cena *elf*: com a configuração 2D fixa, varia-se apenas a leitura do *cache* (desligado, leitura por vizinho mais próximo e leitura trilinear, com $n = 3$ por variante e mesmo *check-*

Tabela 4 – Parte 2 na forma configuração 2D (Parte 1, sem *voxel cache*) → configuração com *voxel cache* (Parte 2, STG 3,0), por cena. Cada célula mostra a *média* sobre $n = 3$ execuções, mesmo *checkpoint* por cena. Trata-se de uma comparação *entre configurações* (a coluna *voxel* inclui o rebalanceamento de STG), e não de um isolamento do *cache*. A coluna EMV_{3D} é a *EditMaskVariance_3D*, métrica de consistência multivista calculada a partir de deltas de RGB renderizado retroprojetados (e não da máscara de treinamento), lida com cautela (menor é melhor apenas nominalmente, pois é confundida pela força de edição; ver a discussão a seguir). Como na Tabela 3, o melhor valor de cada célula aparece em verde; como as duas configurações estão muito próximas, só recebem destaque as diferenças que excedem o desvio-padrão combinado das sementes ($n = 3$). Apenas cinco células se separam do ruído: três a favor do *voxel cache* (*stormtrooper* em *CLIP Direction*; *Einstein* e *fangzhou* em LPIPS) e duas a favor da configuração 2D (*bear* em *CLIP Direction*; *Einstein* em *MV Cos*), o que reforça o caráter neutro do módulo no agregado. Os valores completos de média $\pm$ desvio constam do Apêndice D.

Cena	$CLIP_{dir} \uparrow$ (2D $\rightarrow$ Vox.)	$CLIP-I \uparrow$ (2D $\rightarrow$ Vox.)	SSIM $\uparrow$ (2D $\rightarrow$ Vox.)	LPIPS $\downarrow$ (2D $\rightarrow$ Vox.)	$MV Cos \uparrow$ (2D $\rightarrow$ Vox.)	$EMV_{3D} \downarrow$ (2D $\rightarrow$ Vox.)
bear	0,174 $\rightarrow$ 0,166	0,788 $\rightarrow$ 0,787	0,768 $\rightarrow$ 0,769	0,234 $\rightarrow$ 0,232	0,896 $\rightarrow$ 0,893	0,0338 $\rightarrow$ 0,0339
face (elf)	0,128 $\rightarrow$ 0,131	0,601 $\rightarrow$ 0,603	0,834 $\rightarrow$ 0,833	0,231 $\rightarrow$ 0,231	0,928 $\rightarrow$ 0,925	0,0243 $\rightarrow$ 0,0237
face (Einstein)	0,078 $\rightarrow$ 0,082	0,559 $\rightarrow$ 0,561	0,806 $\rightarrow$ 0,810	0,258 $\rightarrow$ 0,246	0,921 $\rightarrow$ 0,916	0,0097 $\rightarrow$ 0,0102
person (clown)	0,143 $\rightarrow$ 0,144	0,772 $\rightarrow$ 0,781	0,935 $\rightarrow$ 0,935	0,198 $\rightarrow$ 0,198	0,910 $\rightarrow$ 0,907	0,0080 $\rightarrow$ 0,0079
person (stormtrooper)	0,171 $\rightarrow$ 0,175	0,682 $\rightarrow$ 0,676	0,839 $\rightarrow$ 0,838	0,295 $\rightarrow$ 0,293	0,901 $\rightarrow$ 0,899	0,0054 $\rightarrow$ 0,0059
fangzhou (helmet)	0,182 $\rightarrow$ 0,189	0,764 $\rightarrow$ 0,758	0,834 $\rightarrow$ 0,838	0,282 $\rightarrow$ 0,276	0,943 $\rightarrow$ 0,943	0,0182 $\rightarrow$ 0,0188
yuseung (Batman)	0,165 $\rightarrow$ 0,162	0,653 $\rightarrow$ 0,651	0,889 $\rightarrow$ 0,888	0,251 $\rightarrow$ 0,252	0,935 $\rightarrow$ 0,934	0,0111 $\rightarrow$ 0,0104
yuseung (Joker)	0,172 $\rightarrow$ 0,172	0,655 $\rightarrow$ 0,662	0,923 $\rightarrow$ 0,922	0,225 $\rightarrow$ 0,225	0,938 $\rightarrow$ 0,935	0,0221 $\rightarrow$ 0,0207
Média	0,152 $\rightarrow$ 0,153	0,684 $\rightarrow$ 0,685	0,854 $\rightarrow$ 0,854	0,247 $\rightarrow$ 0,244	0,922 $\rightarrow$ 0,919	0,0166 $\rightarrow$ 0,0164

Fonte: Produção do autor.

point). A editabilidade permanece no piso de ruído nas três variantes (*CLIP Direction* 0,128 / 0,131 / 0,127), enquanto a consistência multivista renderizada é maior com a leitura trilinear: a *MV Pairwise Cos Sim* passa de 0,928 (sem *cache*) a 0,933 com trilinear (contra $\approx$ 0,924 com vizinho mais próximo), e a *EditMaskVariance_3D* cai de 0,0243 para $\approx$ 0,0225. A leitura trilinear é o componente decisivo: é ela que converte a consistência no nível da máscara em consistência na imagem renderizada, removendo o falso desacordo introduzido pela discretização do *cache*. A Figura 22 ilustra o efeito em duas vistas. O ganho é pequeno (da ordem do piso de ruído) e, na comparação de configurações da Tabela 4, em que a redução de STG atua em sentido contrário, ele não se traduz em aumento líquido de *MV Cos*; ainda assim, é a evidência mais limpa do mecanismo pelo qual o *cache* promove coerência entre vistas.

A métrica *EditMaskVariance_3D* (Capítulo 5) deve ser lida com cautela: é um *proxy post-hoc* calculado a partir de deltas de RGB renderizado, *desacoplado* da máscara usada no treinamento, e é confundida pela força de edição (a sobre-edição reduz sua variância sem ganho real de consistência), de modo que “menor = melhor” não se sustenta isoladamente. Por isso reporta-se também a *MV Pairwise Cos Sim*, no nível da imagem renderizada, e as duas métricas são lidas em conjunto com a

Figura 22 – Consistência multivista renderizada na cena *elf* sob duas estratégias de leitura do *voxel cache*. Na linha superior (vizinho mais próximo), a discretização em blocos do *cache* introduz falso desacordo entre as vistas; na linha inferior (trilinear ponderada por observação) a mesma região fica coerente entre os ângulos. A leitura trilinear eleva a *MV Pairwise Cos Sim* de $\approx 0,924$ para 0,933.



Vizinho mais próximo — vista 1



Vizinho mais próximo — vista 2



Trilinear — vista 1



Trilinear — vista 2

Fonte: Produção do autor.

editabilidade.

6.3.4 Cena de recoloração ampla (*clown*): o regime de amplificação

Na cena *clown*, o *cache* expõe o segundo regime: **consistente não significa desejado**. O processo 2D pinta ocasionalmente os braços de vermelho mesmo sem o *cache* (em cerca de 1/3 das execuções); como o modelo deposita força de edição moderada e *coerente* nos braços, uma fusão positiva forte amplifica exatamente esse sinal: com peso de mistura dobrado ($\alpha_{\max}=0,4$) a taxa de falha sobe para 2/3. O mapa de variância de depuração (Figura 15) mostra a cabeça escura (edição desejada, baixa variância entre vistas) e os braços um pouco mais claros (variância mais alta): a região propensa à sobre-edição é, ainda que com contraste modesto, *distinguível por variância*. Com o pacote ajustado a esse regime ($\alpha_{\max}=0,2$, $\sigma_{\max}^2$ posicionado no vão

entre cabeça e braços, *warmup* tardio), a taxa de falha retorna ao patamar base de $1/3$ ($n = 3$), isto é, a mesma taxa observada sem o *cache*. A falha residual pertence, portanto, ao processo 2D, e não ao *cache*: por construção, uma fusão apenas positiva só pode *adicionar* força concordante entre vistas, jamais *remover* uma sobre-edição que o processo 2D já produz; de forma coerente, a taxa de falha com o *cache* iguala a da configuração sem ele.

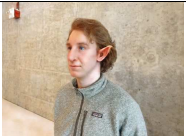
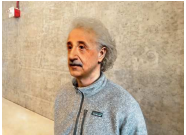
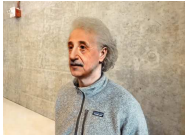
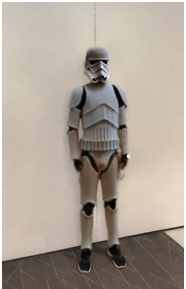
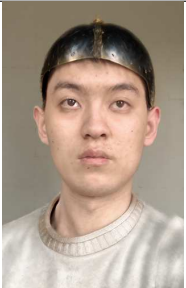
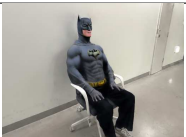
Mais importante que os números, há um problema que o *voxel cache resolve*: a inconsistência entre vistas, que foi a motivação original do módulo. Ela foi observada em uma das três execuções da configuração DreamCatalyst base, o que confirma que a vulnerabilidade não é um artefato da configuração proposta, e sim um modo de falha do próprio processo de supervisão 2D vista a vista. Sem o *cache*, a falha podia ser inconsistente entre câmeras: o braço vermelho visto de um ângulo e com tom de pele visto de outro, o modo de falha da Figura 16, em que a mesma região física recebe decisões opostas conforme a vista. Com o *cache*, sob inspeção visual, as falhas observadas tornam-se *coerentes em 3D*: quando os braços são pintados, são pintados em *todas* as vistas; quando não são, em nenhuma. A estocasticidade migra, portanto, do eixo das *vistas* para o eixo das *execuções*: cada execução passa a ter um desfecho binário e internamente consistente, em vez de um desfecho misto dentro da mesma cena. Esse é exatamente o papel para o qual o módulo foi concebido, e ele o cumpre: um artefato coerente entre vistas é *preferível* a um que muda com o ângulo da câmera (e mais fácil de detectar e descartar por semente). Não é, porém, o ideal: o objetivo final continua sendo reduzir a *taxa* do desfecho indesejado, que permanece no patamar do processo 2D: o limite estrutural discutido a seguir.

Essa assimetria estrutural motivou um experimento exploratório com um ramo de *supressão ativa de regiões contestadas*, que usaria a própria discordância entre vistas como sinal de contenção da edição. O mecanismo foi avaliado, mas **descartado na configuração final**. Com ganho alto, a supressão na verdade *causava* a pintura dos braços (três de três execuções, contra a taxa base de $1/3$), por um laço de **auto-extinção**: ao manter os braços uniformes entre vistas, a supressão extingue a própria variância que a dispara; a contestação desaparece e o ramo positivo reabre, recriando o modo de falha original. Com ganho moderado, o dano desaparecia, mas a taxa de falha ficava *estatisticamente indistinguível* da taxa base do processo 2D. A conclusão é um **teto de estabilidade**: qualquer ganho que o laço de realimentação tolera é insuficiente para reduzir a taxa de falha. Registra-se, para trabalho futuro, um risco adicional: geometria *emergente* (como o capacete do *stormtrooper*, ausente na cena fonte) tende a acumular a assinatura de região contestada (observações suficientes e variância alta), de modo que a supressão ancoraria estrutura nova em direção ao fundo da cena.

6.3.5 Resultados qualitativos da Parte 2

A Figura 23 compara, por cena, a reconstrução fonte, a configuração universal (Parte 1, com o *voxel cache* desligado) e a configuração com *voxel cache* (Parte 2). A comparação é entre *configurações*: a da Parte 2 difere pela presença do *cache* e pelo rebalanceamento de STG para 3,0 (Seção 6.1.1). Os números por cena estão na Tabela 4, e o isolamento controlado do efeito do *cache* na cena *elf*, na Seção 6.3.3.

Figura 23 – Comparação qualitativa da Parte 2 por cena. Da esquerda para a direita: a reconstrução fonte, a configuração universal (Parte 1, sem *voxel cache*) e a configuração com *voxel cache* (Parte 2). O ganho do *cache* é de coerência *entre vistas* (Seção 6.3.4), não de aparência em uma vista isolada.

Edição	Fonte	Parte 1 (sem <i>cache</i>)	Parte 2 (com <i>voxel cache</i>)
Tolkien elf			
			
Clown			
			
Panda			
Helmet armor			
Batman			
			

Fonte: Produção do autor.

6.3.6 Lições sobre os mecanismos da Parte 2

Três lições resumem a Parte 2. Primeiro, a lição *onde vs. para o quê*, agora refinada: uma máscara de localização controla primariamente *onde* o gradiente DDS atua; influenciar *para o que* a edição converge entre vistas exige ou suavizar o próprio valor lido (a leitura trilinear, ligada ao pequeno ganho de *MV Cos* da *elf* no isolamento controlado), ou usar as estatísticas 3D como sinal ativo, caminho cujo teto de estabilidade o experimento de supressão delimitou empiricamente. Segundo, *abstenção não é proteção*: todos os ramos condicionados por confiança se abstêm exatamente onde o *gate* desconfia, de modo que regiões sobre-editadas coerentes ficam entregues ao processo 2D. Terceiro, *o gate de concordância é autorreferente*: a variância que ele lê é alterada pelas próprias intervenções que ele autoriza, o que limita o ganho de qualquer mecanismo realimentado por ela: a autoextinção observada no experimento de supressão é o exemplo medido. Essas lições orientam o trabalho futuro ao nível do *sinal* (*guidance* consistente entre vistas ou um campo 3D semântico/de aparência aprendido, no estilo *Semantic NeRF* (Zhi et al., 2021)), como complemento, e não substituto, da memória 3D aqui caracterizada.

6.4 ANÁLISE DOS MECANISMOS DO MÉTODO

Esta seção discute *por que* certas escolhas funcionam em algumas edições e não em outras. Mais do que tabelas, essas observações explicam o comportamento do método e orientam o uso de cada módulo.

6.4.1 Trajetórias monotônicas vs não monotônicas

As edições diferem na forma como o latente evolui ao longo do treinamento. Em uma edição *monotônica* (por exemplo, *person* $\rightarrow$ *stormtrooper*), cada passo empurra o latente aproximadamente na mesma direção: o modelo pinta a armadura e o capacete de forma cumulativa. Nesse caso, um escalonamento de STG concentrado no início (*decay*) explora a fase de empurrão consistente e produz boa qualidade estrutural.

Em uma edição *não monotônica* (por exemplo, *person* $\rightarrow$ *clown*), a trajetória tem um excesso transitório: o modelo primeiro pinta amplamente (inclusive regiões que deveriam permanecer com aparência de pele) e só depois refina, removendo a tinta onde não é necessária. Amplificar a fase intermediária com STG forte tende a fixar o estado de excesso, dificultando o refinamento posterior. Esse contraste explica por que nenhuma forma única de escalonamento de STG é ótima para todas as cenas simultaneamente: a forma da trajetória, não o instante de treinamento, determina quando a amplificação ajuda.

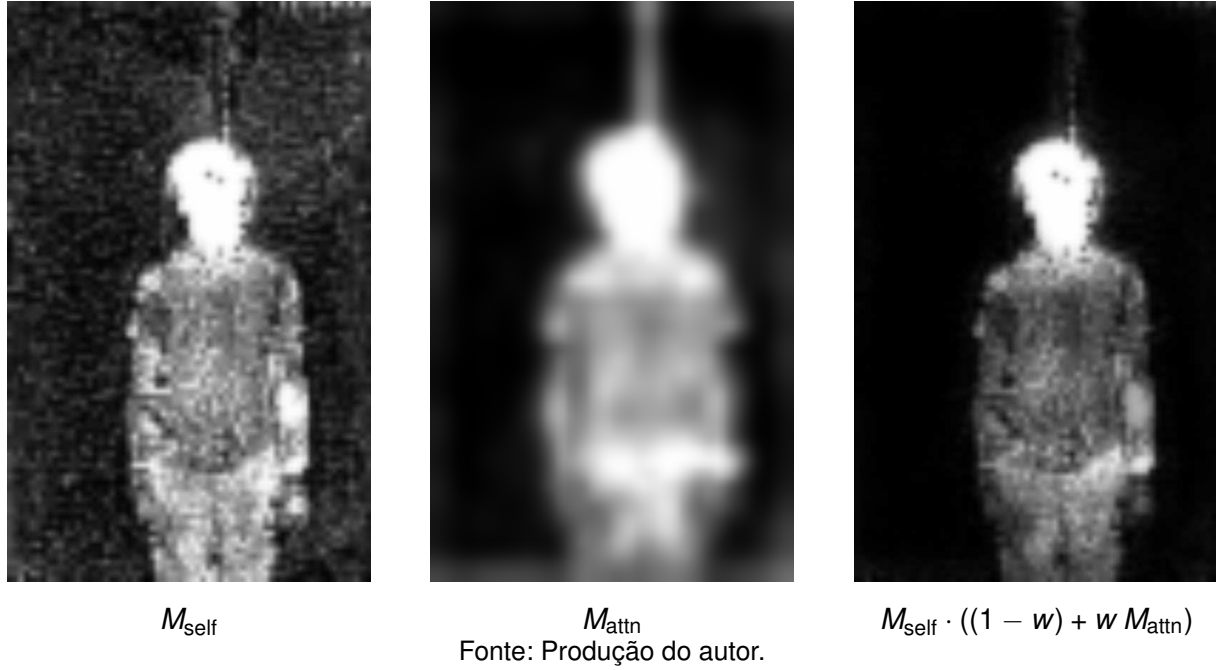
A escala global de STG modula esse mesmo compromisso, e o efeito de reduzi-la é dependente da cena. Reduzindo a escala de 3,5 para 3,0, observou-se, nas exe-

cuções realizadas, que a edição *clown* passou a produzir menos casos com ambos os braços pintados: a menor amplificação contém o excesso transitório da trajetória não monotônica. A mesma redução, porém, tem custo simétrico nas edições que dependem de *completar* estrutura fina: na cena *bear* $\rightarrow$ *panda*, ela degradou a editabilidade, com as patas do panda deixando de ser pintadas, pois ali é justamente a força de edição amplificada que conclui a recoloração. Esse antagonismo, em que menos amplificação ajuda a conter o excesso da *clown*, mas subedita a *bear*, confirma que a escala de STG, assim como o escalonamento, não admite um valor único ótimo entre cenas. A configuração com *voxel cache* adota 3,0 (Seção 6.1.1): a pequena perda de editabilidade na *bear*, dentro do piso de ruído no agregado (Tabela 4), é aceita em troca da contenção da sobre-edição nas cenas de recoloração ampla.

6.4.2 A máscara de *Cross-Attention* é subtrativa por construção

A fusão da máscara semântica com a máscara auto-derivada tem a forma $M = M_{\text{self}} \cdot ((1 - w) + w M_{\text{attn}})$, com $M_{\text{attn}} \in [0, 1]$. Como o fator $((1 - w) + w M_{\text{attn}}) \leq 1$, a fusão só pode *reduzir* (ou manter) a máscara auto-derivada, nunca aumentá-la. Em consequência, a máscara de *Cross-Attention* contribui de forma útil apenas quando a atenção do *token*-alvo é espacialmente nítida (por exemplo, *helmet*, em que a atenção se concentra em uma região definida e suprime o restante). Para *tokens* difusos de estilo ou identidade, cuja atenção é espalhada e de valor intermediário sobre toda a região editada, a fusão apenas atenua uniformemente a máscara, enfraquecendo a edição sem agregar localização. A Figura 24 ilustra a diferença entre a máscara auto-derivada (nítida) e a máscara de *Cross-Attention* (mais difusa) na cena *clown*.

Figura 24 – Máscaras na cena *clown*: a máscara auto-derivada (M_{self}) é nítida sobre o sujeito, enquanto a máscara de *Cross-Attention* (M_{attn}) é mais difusa. A máscara fundida resulta da modulação multiplicativa, que nunca excede a máscara auto-derivada.



6.4.3 A força de edição como envelope de decaimento implícito

A força de edição $s = \|\epsilon_{\text{tgt}} - \epsilon_{\text{src}}\| / (\|\epsilon_{\text{tgt}}\| + \|\epsilon_{\text{src}}\|)$ cresce monotonicamente ao longo do treinamento, à medida que a cena renderizada se afasta da fonte. Como tanto o STG quanto a âncora de fundo são multiplicados por $(1 - s)$, a força de edição funciona como um *envelope de decaimento adicional* aplicado por cima dos escalonamentos temporais explícitos: os módulos se atenuam não apenas em função do passo de difusão, mas também em função de quanto a cena já foi editada. Esse envelope é, por construção, adaptativo à cena (cenar que divergem mais da fonte atenuam mais cedo), mas se compõe com os escalonamentos explícitos, e essa composição precisa ser levada em conta ao interpretar o comportamento temporal do método.

6.4.4 Balanço entre força de edição e preservação

O gradiente combina um termo de edição, $w_{\text{DDS}} (\epsilon_{\text{tgt}}^{\text{loc}} - \epsilon_{\text{src}})$, e um termo de preservação, $\exp(t_{\text{norm}}) \psi(x_0^{\text{tgt}} - x_0^{\text{src}})$, que puxa o latente de volta à fonte. O coeficiente de preservação *diminui* no fim do treinamento (pois $\exp(t_{\text{norm}})$ decresce com t_{norm}). Assim, quando uma edição enfraquece no fim do treinamento, a causa não é o crescimento da preservação, e sim o *colapso da força de edição*: quando os amplificadores (STG, TAG) se atenuam e uma máscara difusa dilui o termo de edição na própria região editada, o termo de preservação, ainda que menor, passa a dominar

localmente. Esse mecanismo explica casos em que uma edição visualmente forte em iterações intermediárias regride em direção à fonte ao final, sem que a preservação tenha aumentado.

6.4.5 Robustez a *guidance* elevado: supressão do clone fantasma

A comparação principal mantém o *guidance scale* fixo em 7,5 por fidelidade ao DreamCatalyst base (Seção 6.1). Um teste de robustez à parte, com *guidance* deliberadamente elevado (12,5), revela um modo de falha do processo base: ao amplificar de forma agressiva o sinal textual, o DreamCatalyst pode *duplicar* o sujeito da edição e criar um segundo personagem na cena, por exemplo um segundo *stormtrooper* ao lado do original (Figura 25a). O efeito surge porque o *guidance* forte espalha a direção de edição para além da região relevante, materializando o conteúdo do *prompt* em pixels que deveriam permanecer como fundo.

Tanto a configuração 2D quanto a com *voxel cache* suprimem esse artefato sob o mesmo *guidance* elevado (Figura 25b,c): a *source-blended localization* e a âncora de preservação fora da máscara confinam a edição à região de fato relevante, de modo que a força textual adicional intensifica a edição pretendida em vez de instanciar uma cópia em outro ponto da imagem. É, portanto, uma evidência de que os módulos de localização e preservação não apenas melhoram a preservação média (Seção 6.2), mas também ampliam a margem de *guidance* utilizável antes que a edição degenere. Esse espaço extra não foi explorado nesta versão (a comparação principal fixa o *guidance* em 7,5), mas abre uma direção concreta de trabalho futuro: com o vazamento contido pelas máscaras, torna-se viável *eleva*r ou reescalonar o *guidance scale*, ou ajustar parâmetros do tipo *FreeU* (Si et al., 2024) (reponderação das conexões de *skip* e do *backbone* da UNet), para buscar mais editabilidade sem o vazamento que normalmente acompanha esses ajustes.

Figura 25 – Robustez a *guidance* elevado na cena *stormtrooper*, com *guidance scale* 12,5 nas três imagens. (a) No DreamCatalyst base, o *guidance* elevado duplica o sujeito e cria um segundo *stormtrooper* (clone fantasma). (b,c) Tanto a configuração 2D quanto a com *voxel cache* confinam a edição e não criam clone, ainda sob o mesmo *guidance* 12,5.



(a) DreamCatalyst base

(b) Configuração 2D proposta
Fonte: Produção do autor.

(c) Com *voxel cache*

6.5 ESCOPO E LIMITAÇÕES

Cinco pontos delimitam a leitura dos resultados: o tipo de edição para o qual o método foi projetado, um artefato específico em edições que encolhem a silhueta do alvo, o número de execuções por configuração, o número e a diversidade de cenas, e a dependência dos *prompts* de edição.

Transferência de estilo global está fora do escopo. A pilha de localização proposta pressupõe uma edição de objeto ou identidade com fundo a ser preservado. Para edições de estilo que devem alterar a imagem inteira (por exemplo, transformar toda a cena no estilo de um pintor), a localização é contraproducente: o *token* de estilo não tem ancoragem espacial nítida, de modo que a máscara semântica apenas atenua a edição (Seção 6.4.2), e o DDS base já realiza esse tipo de edição global de forma adequada. Trata-se, portanto, de uma fronteira de escopo justificada, e não de uma deficiência: o método tem como alvo edições localizadas de objeto e identidade, regime em que a localização melhora a preservação do fundo. Edições de estilo global permanecem como uma extensão natural para trabalhos futuros, possivelmente com um seletor que desligue a localização quando a edição não for espacialmente localizável.

Edições que encolhem a silhueta: o efeito de “mão fantasma”. Na cena *stormtrooper*, a mão enluvada do *stormtrooper* é menor que a mão original da pessoa.

Parte da região antes ocupada pela mão humana deixa, então, de ser coberta pela nova silhueta e deveria ser *apagada* (substituída por fundo), e não reestilizada. Essa exigência é o oposto do que a pilha de localização e preservação foi projetada para fazer: fora da máscara de edição, a âncora de preservação puxa a região de volta à fonte, de modo que o resíduo da mão original é *conservado* em vez de removido, um artefato translúcido de “mão fantasma”. Por inspeção visual, ele aparece de forma mais marcante na configuração proposta do que no DreamCatalyst base, cuja localização mais fraca edita a região com mais liberdade; o artefato em si, porém, é estrutural ao regime de formação do *stormtrooper*, e não um defeito de ajuste. Trata-se do reverso direto do ganho de preservação de fundo: a mesma localização que protege regiões fora da edição penaliza as edições de “espaço negativo”, que exigem encolher ou remover estrutura da cena fonte. O método é, portanto, mais forte quando o alvo *ocupa* a região editada do que quando a edição precisa *esvaziá-la*.

Número de execuções. Os resultados usam $n = 3$ execuções por configuração. Esse número estabelece médias e desvios mais confiáveis do que uma única execução (importante porque a variação de execução para execução neste *pipeline* é da ordem de 0,005 a 0,01 em *CLIP Direction*, Seção 6.1), mas ainda é pequeno para estimativas precisas de variância. Por isso, conclusões marginais (diferenças próximas ao piso de ruído) são explicitamente sinalizadas como inconclusivas e não recebem destaque na Tabela 3. A ampliação do número de execuções nas cenas principais fica registrada como trabalho de continuidade.

Número e diversidade de cenas. A avaliação cobre oito edições sobre um conjunto reduzido de cenas, escolhidas para representar eixos distintos de dificuldade (edições de identidade, de corpo inteiro, com fundo visível e com risco de inconsistência entre vistas). Esse conjunto é suficiente para caracterizar tendências e regimes de comportamento, mas é pequeno para afirmar generalização ampla a outras cenas, objetos e categorias de edição: as conclusões devem ser lidas como evidência consistente nesse conjunto, e não como garantia universal.

Dependência dos *prompts*. Como a edição é condicionada por texto, os resultados dependem das instruções de fonte e alvo efetivamente avaliadas (Tabela 5). Reformulações de *prompt* podem deslocar tanto a força quanto a localização da edição e, com isso, o equilíbrio entre editabilidade e preservação. A sensibilidade a essas reformulações não foi medida de forma sistemática e fica registrada como trabalho de continuidade.

O Capítulo 7 sintetiza estas observações, consolida as contribuições de cada parte e delimita as direções de trabalho futuro.

7 CONCLUSÃO

Esta monografia apresentou a organização conceitual, metodológica e experimental do projeto. O trabalho parte do DreamCatalyst e propõe extensões para tornar a edição 3D orientada por texto mais localizada, preservadora e coerente entre vistas. A contribuição não está na criação de um novo modelo de difusão ou de uma nova representação 3D, mas na forma como os sinais de difusão são tratados antes de atualizar a cena.

7.1 SÍNTESE DO TRABALHO

O problema estudado é a edição de cenas 3D reconstruídas a partir de múltiplas imagens usando comandos textuais. O método base, DreamCatalyst, já permite otimizar representações 3D com supervisão de um modelo de difusão 2D. Entretanto, essa abordagem pode editar regiões indesejadas, alterar o fundo ou produzir decisões inconsistentes entre vistas.

Para atacar essas limitações, este trabalho organiza uma extensão modular composta por TAG/STG, máscaras auto-derivadas, máscaras semânticas de *Cross-Attention* e *source-blended localization*. A extensão também inclui âncoras de preservação (uma fora da máscara e uma simples de média latente), adaptação por força de edição (*edit strength*), um *voxel cache* 3D de evidências de edição e uma métrica dedicada de consistência multivista (*EditMaskVariance_3D*), cuja avaliação (Parte 2) caracterizou um regime de utilidade por cena para a consistência multivista. Cada componente foi projetado para poder ser avaliado separadamente e desligado, de modo a reproduzir o método base.

7.2 SÍNTESE DOS RESULTADOS

A campanha experimental consolidou os resultados apresentados no Capítulo 6. Na Parte 1, que constitui o resultado central do trabalho, a configuração universal proposta preservou melhor as regiões fora da edição, mantendo uma editabilidade equivalente à do DreamCatalyst base (medida por *CLIP Direction*, $0,151 \rightarrow 0,152$ na média): a proposta melhora o SSIM nas oito edições e reduz o LPIPS em sete delas (em média, SSIM $0,825 \rightarrow 0,854$, LPIPS $0,272 \rightarrow 0,247$ e *CLIP Image Sim* $0,664 \rightarrow 0,684$). Esse é o suporte quantitativo mais forte do trabalho e confirma o compromisso entre força de edição e preservação como o eixo central do método. A Parte 2 investigou o *voxel cache* 3D como módulo opcional de consistência entre vistas e caracterizou um regime de utilidade por cena. A formulação inicial, que agregava a própria máscara 2D, não reduzia a variância multivista: o teste A/B contra um controle apenas-medição foi nulo. Esse achado motivou o pacote final de *força invariante à*

vista. Com esse pacote, contra a configuração base correta (mesma configuração 2D, *cache* desligado), o módulo cumpre a função para a qual foi concebido. A decisão de edição passa a ser compartilhada pelas câmeras, e o modo de falha que motivou o módulo, a mesma região editada em uma vista e preservada em outra, dá lugar a desfechos coerentes em 3D, deslocando a estocasticidade do ângulo de observação para a execução. Nas métricas agregadas o módulo é neutro, todas no piso de ruído, e seu ganho de consistência multivista aparece apenas em isolamento controlado e por margem pequena. Em cenas de recoloração ampla, o módulo requer o *gate* de concordância por variância para não amplificar sinais coerentes e indesejados, e a taxa do desfecho indesejado permanece no patamar do processo 2D. A tentativa de reduzi-la por supressão ativa de regiões contestadas encontrou um teto de estabilidade abaixo de qualquer ganho útil e foi desativada. A lição estrutural se mantém, agora refinada: a máscara de localização controla primariamente *onde* o gradiente DDS atua; influenciar *para o que* a edição converge exige suavizar o valor lido do *cache* (leitura trilinear) ou usar as estatísticas 3D como sinal ativo, caminho cujo limite o experimento de supressão delimitou.

7.3 ESCOPO E LIMITAÇÕES OBSERVADAS

Alguns pontos delimitam o escopo atual do trabalho e devem ser considerados na interpretação dos resultados:

- o método ainda depende da qualidade da reconstrução inicial e das poses de câmera estimadas;
- as máscaras derivadas de difusão podem falhar quando o *prompt* é ambíguo ou quando o modelo 2D não associa corretamente o *token* à região desejada;
- o *voxel cache* 3D é um módulo opcional cujo benefício depende do regime da cena: nas oito edições ($n = 3$) é neutro nas métricas agregadas, com um ganho de consistência renderizada pequeno e restrito ao isolamento controlado (cena de detalhe fino), enquanto em recoloração ampla exige ajuste do *gate* de concordância por cena;
- o número de execuções por configuração ($n = 3$) e o conjunto reduzido de cenas e *prompts* avaliados limitam a análise estatística e a generalização das conclusões a outras cenas e instruções;
- métricas automáticas podem não capturar todos os artefatos percebidos visualmente;
- uma configuração universal pode não ser ótima para todos os tipos de edição.

7.4 TRABALHOS FUTUROS

As principais direções futuras são:

- consolidar a Parte 2 nas cenas pendentes e complementar a memória 3D de evidências com mecanismos no nível do *signal: guidance* consistente entre vistas (*cross-view*) ou um campo 3D semântico/de aparência treinável, inspirado em *Semantic NeRF*;
- integrar priores (*priors*) de segmentação referencial ou *grounding* multimodal para melhorar a localização semântica;
- projetar perdas explícitas de consistência multivista;
- estender a avaliação para vistas separadas para teste (*held-out views*), em vez de usar apenas vistas de treinamento;
- avaliar sistematicamente a transferência dos módulos para *3D Gaussian Splatting*;
- estudar mecanismos de preservação geométrica baseados em profundidade;
- explorar a margem extra de *guidance* aberta pela contenção de vazamento das máscaras (Seção 6.4.5): elevar ou reescalonar o *guidance scale* e ajustar parâmetros do tipo *FreeU* (Si et al., 2024) em busca de mais editabilidade, agora que o vazamento que normalmente acompanha esses ajustes é suprimido.

7.5 CONSIDERAÇÕES FINAIS

O trabalho propõe uma leitura prática da edição 3D por difusão: o desafio não é apenas gerar uma imagem semanticamente correta em uma vista, mas controlar onde o gradiente atua, quanto da cena original deve ser preservado e como decisões de edição são compartilhadas entre múltiplas câmeras. Os resultados consolidados no Capítulo 6 sustentam essa leitura, separando a contribuição universal de *guidance* (Parte 1, com suporte quantitativo) da análise da Parte 2, cujo resultado é um regime de utilidade por cena para a memória 3D de evidências: ela melhora a consistência renderizada quando o desacordo entre vistas é ruído de detalhe fino, e precisa de *gates* de concordância (ou da própria discordância usada como sinal de contenção) quando o modelo deposita força coerente em regiões que deveriam ser preservadas.

REFERÊNCIAS

BRACK, Manuel; FRIEDRICH, Felix; HINTERSDORF, Dominik; STRUPPEK, Lukas; SCHRAMOWSKI, Patrick; KERSTING, Kristian. LEDITS++: Limitless Image Editing using Text-to-Image Models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2024. Disponível em: <https://arxiv.org/abs/2311.16711>.

BROOKS, Tim; HOLYNSKI, Aleksander; EFROS, Alexei A. InstructPix2Pix: Learning to Follow Image Editing Instructions. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2211.09800>.

CHO, Myeongjin et al. Tangential Amplified Guidance. **arXiv preprint arXiv:2510.04533**, 2025.

COUAIRON, Guillaume; VERBEEK, Jakob; SCHWENK, Holger; CORD, Matthieu. DiffEdit: Diffusion-based Semantic Image Editing with Mask Guidance. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2210.11427>.

FINCH, Tony. Incremental Calculation of Weighted Mean and Variance. **University of Cambridge Computing Service, Technical note**, 2009.

FISHER, Ronald A. Dispersion on a Sphere. **Proceedings of the Royal Society of London. Series A**, v. 217, n. 1130, 1953.

HERTZ, Amir; MOKADY, Ron; TENENBAUM, Jay; ABERMAN, Kfir; PRITCH, Yael; COHEN-OR, Daniel. Delta Denoising Score. In: IEEE/CVF International Conference on Computer Vision. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2304.07090>.

HERTZ, Amir; MOKADY, Ron; TENENBAUM, Jay; ABERMAN, Kfir; PRITCH, Yael; COHEN-OR, Daniel. Prompt-to-Prompt Image Editing with Cross Attention Control. **arXiv preprint arXiv:2208.01626**, 2022. Disponível em: <https://arxiv.org/abs/2208.01626>.

HO, Jonathan; JAIN, Ajay; ABBEEL, Pieter. Denoising Diffusion Probabilistic Models. In: ADVANCES in Neural Information Processing Systems. [S.l.: s.n.], 2020. Disponível em: <https://arxiv.org/abs/2006.11239>.

HO, Jonathan; SALIMANS, Tim. Classifier-Free Diffusion Guidance. **arXiv preprint arXiv:2207.12598**, 2022. Disponível em: <https://arxiv.org/abs/2207.12598>.

HYUNG, Junha et al. Spatiotemporal Skip Guidance for Enhanced Video Diffusion Sampling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2025.

KAJIYA, James T.; VON HERZEN, Brian P. Ray Tracing Volume Densities. **ACM SIGGRAPH Computer Graphics**, v. 18, n. 3, 1984.

KERBL, Bernhard; KOPANAS, Georgios; LEIMKUEHLER, Thomas; DRETTAKIS, George. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. **ACM Transactions on Graphics**, 2023. Disponível em: <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>.

MILDENHALL, Ben; SRINIVASAN, Pratul P.; TANCIK, Matthew; BARRON, Jonathan T.; RAMAMOORTHY, Ravi; NG, Ren. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: EUROPEAN Conference on Computer Vision. [S.l.: s.n.], 2020. Disponível em: <https://arxiv.org/abs/2003.08934>.

NERFSTUDIO TEAM. **Nerfstudio Documentation: NeRF – Neural Radiance Fields**. 2023. Disponível em: <https://docs.nerf.studio/nerfology/methods/nerf.html>. Acesso em: 11 jun. 2026.

POOLE, Ben; JAIN, Ajay; BARRON, Jonathan T.; MILDENHALL, Ben. DreamFusion: Text-to-3D using 2D Diffusion. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2209.14988>.

SCHOENBERGER, Johannes L.; FRAHM, Jan-Michael. Structure-from-Motion Revisited. In: IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2016. Disponível em: <https://demuc.de/papers/schoenberger2016sfm.pdf>.

SHIM, Hyunjung et al. DreamCatalyst: Fast and High-Quality 3D Editing via Controlling Editability and Identity Preservation. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2025. Disponível em: <https://arxiv.org/abs/2407.11394>.

SI, Chenyang; HUANG, Ziqi; JIANG, Yuming; LIU, Ziwei. FreeU: Free Lunch in Diffusion U-Net. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2024. Disponível em: <https://arxiv.org/abs/2309.11497>.

SIDDIQUI, Yawar; PORZI, Lorenzo; BULO, Samuel Rota; MUELLER, Norman; NIESSNER, Matthias; DAI, Angela; KONTSCHIEDER, Peter. Panoptic Lifting for 3D Scene Understanding with Neural Fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2212.09802>.

TANCIK, Matthew et al. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In: ACM SIGGRAPH 2023 Conference Proceedings. [S.l.: s.n.], 2023. Disponível em: <https://arxiv.org/abs/2302.04264>.

TANG, Raphael; PANDEY, Akshat; JIANG, Zhiying; YANG, Gefei; KUMAR, Karun; LIN, Jimmy; TURE, Ferhan. What the DAAM: Interpreting Stable Diffusion Using Cross Attention. **arXiv preprint arXiv:2210.04885**, 2022. Disponível em: <https://arxiv.org/abs/2210.04885>.

WEN, Mingrui et al. InterGSEdit: Inter-View Consistent 3D Gaussian Splatting Editing. **arXiv preprint arXiv:2507.04961**, 2025. Disponível em: <https://arxiv.org/abs/2507.04961>.

ZHI, Shuaifeng; LAIDLOW, Tristan; LEUTENEGGER, Stefan; DAVISON, Andrew J. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In: IEEE/CVF International Conference on Computer Vision. [S.l.: s.n.], 2021. Disponível em: <https://arxiv.org/abs/2103.15875>.

APÊNDICE A – DETALHES DE REPRODUTIBILIDADE DOS EXPERIMENTOS

Este apêndice reúne as informações necessárias para reproduzir os experimentos. As cenas e as respectivas instruções de edição encontram-se na Tabela 1 (Capítulo 6). Os valores dos parâmetros ativos do método estão no arquivo `method_config.py` do repositório, em dois blocos: `DC_CUSTOM_PARAMS`, para o *guidance*, a localização e a preservação (Parte 1), e `VOXEL_CACHE_PARAMS`, para o *voxel cache* (Parte 2).

Ambiente de software e hardware. Todos os experimentos (reconstrução, edição e avaliação) foram executados em uma única GPU NVIDIA H100, com Python 3.10, PyTorch 2.1.2 (CUDA 11.8), Nerfstudio 1.0.2, *diffusers* 0.27.2 e COLMAP 3.9.1. A edição utiliza o modelo `timbrooks/instruct-pix2pix` e o refinamento utiliza `runwayml/stable-diffusion-v1-5`.

Protocolo comum. Cada configuração é executada com $n = 3$ execuções independentes por edição, sobre o mesmo *checkpoint* de reconstrução *nerfacto* por cena, mantido fixo entre as configurações comparadas. O *guidance scale* permanece fixo em 7,5, conforme o protocolo do Capítulo 5; a edição usa 3000 iterações; e o fator de escala (*downscale*) das imagens é mantido igual na reconstrução, na edição e na avaliação.

A escala de STG difere entre as configurações: 3,5 na configuração universal (Parte 1, sem *cache*) e 3,0 na configuração com *voxel cache* (Parte 2). No prefixo da Parte 2, esse valor e os demais aparecem no sufixo: STG3 (STG 3,0), MB02 (*max_blend* 0,2) e maxVar02 (*max_variance* 0,02).

APÊNDICE B – PROMPTS DE EDIÇÃO: FONTE E ALVO

A Tabela 1 (Capítulo 6) apresenta a instrução de edição de cada cena, ou seja, o *prompt* alvo (t_{tgt}), mas não o *prompt* fonte (t_{src}). Este apêndice reúne ambos em um único lugar, a fim de facilitar a reprodução. Ambos são passados à etapa de edição por meio de `scripts/edit.sh` (na forma `<src_prompt> <tgt_prompt>`): o *prompt* fonte condiciona o ramo *source* do DDS (ϵ_{src}) e o *prompt* alvo condiciona o ramo *target* (ϵ_{tgt}). Conforme o protocolo adotado, o *prompt* fonte é uma descrição neutra da cena original, ao passo que o *prompt* alvo é a instrução de edição (no estilo do InstructPix2Pix), que especifica apenas a alteração desejada.

Tabela 5 – Prompts de fonte (t_{src}) e de alvo (t_{tgt}) usados em cada edição.

Edição	Prompt fonte (t_{src})	Prompt alvo (t_{tgt})
bear → panda	a photo of a bear statue	<i>Turn the bear statue into a panda</i>
face → elf	a photo of a face	<i>Turn him into a Tolkien elf</i>
face → Einstein	a photo of a face	<i>Turn him into an Einstein</i>
person → clown	a photo of a person	<i>Turn him into a clown</i>
person → stormtrooper	a photo of a person	<i>Turn him into a stormtrooper</i>
fangzhou → helmet	a photo of a face	<i>Put a knight's helmet on him</i>
yuseung → Batman	a photo of a man	<i>Turn him into a Batman</i>
yuseung → Joker	a photo of a man	<i>Turn him into a Joker</i>

Fonte: Produção do autor.

APÊNDICE C – RESULTADOS DETALHADOS DA PARTE 1 (MÉDIA E DESVIO PADRÃO)

Este apêndice complementa a Tabela 3 com os valores completos de média $\pm$ desvio padrão sobre as $n = 3$ execuções por cena, para o DreamCatalyst base (Tabela 6) e para a configuração universal proposta (Tabela 7). O desvio padrão é o desvio amostral (denominador $n - 1$) entre as três execuções e fundamenta o critério de significância adotado no Capítulo 6: uma diferença entre configurações só é tratada como real quando excede a dispersão combinada. As setas de cada métrica indicam o sentido de melhora ($\uparrow$ maior é melhor; $\downarrow$ menor é melhor).

Tabela 6 – DreamCatalyst base (DC): média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.

Cena	<i>CLIP</i>_{dir} $\uparrow$	<i>CLIP-I</i> $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	<i>MV Cos</i> $\uparrow$
bear	0,180 $\pm$ 0,006	0,745 $\pm$ 0,003	0,693 $\pm$ 0,006	0,283 $\pm$ 0,003	0,907 $\pm$ 0,003
face (elf)	0,123 $\pm$ 0,003	0,592 $\pm$ 0,006	0,820 $\pm$ 0,002	0,256 $\pm$ 0,006	0,920 $\pm$ 0,001
face (Einstein)	0,081 $\pm$ 0,002	0,557 $\pm$ 0,009	0,794 $\pm$ 0,002	0,255 $\pm$ 0,002	0,922 $\pm$ 0,006
person (clown)	0,138 $\pm$ 0,002	0,773 $\pm$ 0,007	0,924 $\pm$ 0,001	0,227 $\pm$ 0,003	0,911 $\pm$ 0,006
person (stormtrooper)	0,155 $\pm$ 0,003	0,650 $\pm$ 0,005	0,795 $\pm$ 0,001	0,331 $\pm$ 0,000	0,926 $\pm$ 0,002
fangzhou (helmet)	0,206 $\pm$ 0,013	0,727 $\pm$ 0,033	0,796 $\pm$ 0,037	0,311 $\pm$ 0,029	0,951 $\pm$ 0,007
yuseung (Batman)	0,156 $\pm$ 0,004	0,627 $\pm$ 0,009	0,869 $\pm$ 0,004	0,273 $\pm$ 0,003	0,930 $\pm$ 0,005
yuseung (Joker)	0,170 $\pm$ 0,004	0,643 $\pm$ 0,008	0,908 $\pm$ 0,000	0,241 $\pm$ 0,002	0,932 $\pm$ 0,004

Fonte: Produção do autor.

Tabela 7 – Configuração universal proposta: média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.

Cena	<i>CLIP</i>_{dir} $\uparrow$	<i>CLIP-I</i> $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	<i>MV Cos</i> $\uparrow$
bear	0,174 $\pm$ 0,002	0,788 $\pm$ 0,007	0,768 $\pm$ 0,003	0,234 $\pm$ 0,004	0,896 $\pm$ 0,002
face (elf)	0,128 $\pm$ 0,003	0,601 $\pm$ 0,005	0,834 $\pm$ 0,001	0,231 $\pm$ 0,003	0,928 $\pm$ 0,004
face (Einstein)	0,078 $\pm$ 0,003	0,559 $\pm$ 0,008	0,806 $\pm$ 0,005	0,258 $\pm$ 0,010	0,921 $\pm$ 0,002
person (clown)	0,143 $\pm$ 0,008	0,772 $\pm$ 0,013	0,935 $\pm$ 0,002	0,198 $\pm$ 0,001	0,910 $\pm$ 0,003
person (stormtrooper)	0,171 $\pm$ 0,004	0,682 $\pm$ 0,013	0,839 $\pm$ 0,001	0,295 $\pm$ 0,001	0,901 $\pm$ 0,012
fangzhou (helmet)	0,182 $\pm$ 0,017	0,764 $\pm$ 0,006	0,834 $\pm$ 0,002	0,282 $\pm$ 0,004	0,943 $\pm$ 0,002
yuseung (Batman)	0,165 $\pm$ 0,002	0,653 $\pm$ 0,002	0,889 $\pm$ 0,002	0,251 $\pm$ 0,003	0,935 $\pm$ 0,002
yuseung (Joker)	0,172 $\pm$ 0,003	0,655 $\pm$ 0,007	0,923 $\pm$ 0,002	0,225 $\pm$ 0,003	0,938 $\pm$ 0,001

Fonte: Produção do autor.

APÊNDICE D – RESULTADOS DETALHADOS DA PARTE 2 (VOXEL CACHE)

Este apêndice complementa a Tabela 4 com os valores completos de média $\pm$ desvio padrão sobre as $n = 3$ execuções por cena para a configuração com *voxel cache* (Parte 2, STG 3,0). A configuração 2D de referência é a proposta da Parte 1, cujos valores detalhados para as cinco métricas de imagem constam da Tabela 7 (Apêndice C). Mantém-se aqui a métrica adicional *EditMaskVariance_3D* (EMV_{3D}), lida com cautela conforme discutido no Capítulo 6. O desvio padrão é o desvio amostral (denominador $n - 1$) entre as três execuções.

Tabela 8 – Configuração com *voxel cache* (Parte 2): média $\pm$ desvio padrão sobre $n = 3$ execuções por cena.

Cena	$CLIP_{dir} \uparrow$	$CLIP-I \uparrow$	$SSIM \uparrow$	$LPIPS \downarrow$	$MV Cos \uparrow$	$EMV_{3D} \downarrow$
bear	$0,166 \pm 0,006$	$0,787 \pm 0,005$	$0,769 \pm 0,002$	$0,232 \pm 0,002$	$0,893 \pm 0,003$	$0,0339 \pm 0,0003$
face (elf)	$0,131 \pm 0,004$	$0,603 \pm 0,003$	$0,833 \pm 0,003$	$0,231 \pm 0,006$	$0,925 \pm 0,003$	$0,0237 \pm 0,0010$
face (Einstein)	$0,082 \pm 0,007$	$0,561 \pm 0,002$	$0,810 \pm 0,002$	$0,246 \pm 0,001$	$0,916 \pm 0,001$	$0,0102 \pm 0,0008$
person (clown)	$0,144 \pm 0,002$	$0,781 \pm 0,008$	$0,935 \pm 0,001$	$0,198 \pm 0,003$	$0,907 \pm 0,004$	$0,0079 \pm 0,0008$
person (stormtrooper)	$0,175 \pm 0,001$	$0,676 \pm 0,002$	$0,838 \pm 0,004$	$0,293 \pm 0,001$	$0,899 \pm 0,005$	$0,0059 \pm 0,0007$
fangzhou (helmet)	$0,189 \pm 0,006$	$0,758 \pm 0,002$	$0,838 \pm 0,000$	$0,276 \pm 0,003$	$0,943 \pm 0,003$	$0,0188 \pm 0,0013$
yuseung (Batman)	$0,162 \pm 0,004$	$0,651 \pm 0,005$	$0,888 \pm 0,003$	$0,252 \pm 0,001$	$0,934 \pm 0,007$	$0,0104 \pm 0,0012$
yuseung (Joker)	$0,172 \pm 0,004$	$0,662 \pm 0,001$	$0,922 \pm 0,002$	$0,225 \pm 0,003$	$0,935 \pm 0,005$	$0,0207 \pm 0,0016$

Fonte: Produção do autor.