# DEPTH-TO-RGB: REPURPOSING A FROZEN DEPTH ESTIMATOR FOR GEOMETRY-GUIDED COMPOSITING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reference-based object compositing inserts or replaces an object using a background image, a reference image, and a 2D compositing mask. These inputs guide appearance and placement but leave the completed scene's geometry implicit, which can distort object structure or alter the surroundings. Our Depth-to-RGB (D2R) framework predicts composite depth for a scene not yet observed in the RGB inputs. It learns reference-conditioned corrections to a frozen depth estimator using encoder features of paired completed scenes as targets. The unchanged decoder maps the corrected representation to the intended scene's depth, which a separately trained renderer holds fixed during RGB synthesis. Under matched architecture and training, encoder-feature supervision reduces OOD Stage-1 AbsRel by 31.4% relative to decoded-depth supervision. We also introduce AnyInsertion++ with paired in-distribution and category-disjoint splits to evaluate generalization beyond compositing training categories. The complete D2R framework leads 12 open-source and 3 closed-source baselines in estimator-derived geometry and photometric quality on both paired splits. On category-disjoint data, D2R reduces AbsRel by 43.7% and improves PSNR by 2.4 dB over the matched RGB baseline. Across three unpaired benchmarks, D2R leads both identity metrics and reduces mean CLIP reference cosine distance by 55% relative to the strongest baseline. We will publicly release code and AnyInsertion++ upon publication.

## 1 INTRODUCTION

Object compositing must preserve a reference object's identity while adapting its shape and placement to a new scene. The background provides scene context, and a 2D mask indicates the target region without specifying the intended composite depth. Figure 1 illustrates the challenge: transferring reference appearance can produce distorted object structure, incorrect placement, or changes to the surrounding scene. These errors motivate a geometric condition for the completed scene, guiding appearance generation with an explicit description of the object and its surroundings.

Most reference-guided object compositing methods infer geometry implicitly during RGB denoising. Some approaches with explicit geometric conditions use depth estimated from an input image (Chen et al., 2024a; Parihar et al., 2025), assembled foreground and background depth maps (Li et al., 2024a), or an RGB proxy rendered from a user-posed 3D object (Gong et al., 2026b). Depth estimated from an input image describes its observed geometry, which does not determine the reference object's depth and shape in the completed composite. Assembling maps relies on rules for placing and rescaling reference depth within the background, while rendering a proxy requires the object's target pose to be specified before RGB generation. Our goal is to predict post-composite depth from the background image, reference image, and 2D compositing mask, jointly estimating the geometry of the reference object and its surroundings.

We call this field *composite depth*: the depth of the entire scene after compositing. Background depth omits the intended object, while reference depth describes it in another scene and viewpoint. Neither map directly describes the object's shape and placement within the target scene. Predicting this field requires combining reference appearance with scene context. We formulate this task as *endogenous geometry provisioning*: inferring this geometric condition from the given inputs.

A pretrained depth estimator maps an observed image's features to depth. *Depth-to-RGB* (D2R) freezes its encoder and decoder while learning reference-conditioned residual corrections to the



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.




Figure 1: **Predicting Composite Depth Before RGB Synthesis.** Depth-to-RGB (D2R; Ours) predicts composite depth (seventh column) and holds it fixed while generating RGB (eighth). D2R preserves reference appearance while matching the target shape and pose; the open- and closed-source baselines instead miss the target geometry, alter object identity, or modify surrounding content.

background representation. Training aligns the resulting representation with encoder features extracted from paired completed scenes, enabling the unchanged decoder to predict composite depth. This geometric condition remains fixed throughout RGB synthesis.

To evaluate geometry and category generalization, we curate AnyInsertion++ from AnyInsertion (Song et al., 2026), expanding paired evaluation from 38 to 426 cases with a category-disjoint split. Paired evaluation compares estimated depth and RGB composites with paired targets. Additionally, we assess reference identity on TF-ICON (Lu et al., 2023), AIComposer (Li et al., 2025), and ComplexCompo (Lu et al., 2026), which collectively cover photographic, cartoon, and artistic backgrounds (Tab. 4). These unpaired benchmarks test whether reference appearance is retained across different visual styles beyond the paired evaluation setting. Our contributions are:

- **Problem.** We address the missing geometric condition in reference-based object compositing: predicting the completed scene's relative depth from a background image, reference image, and 2D compositing mask.
- **Solution.** We propose *Depth-to-RGB* (D2R). First, we learn reference-conditioned corrections to a frozen depth estimator using encoder features of paired completed scenes as targets. Under matched architecture and training, encoder-feature supervision reduces OOD Stage-1 AbsRel by 31.4% relative to decoded-depth supervision. Second, we train an RGB renderer to use the resulting depth as a fixed geometric condition.
- **Results.** The complete D2R system reduces category-disjoint AbsRel by 43.7% and improves PSNR by 2.4 dB over the matched RGB baseline. Across three unpaired benchmarks, D2R leads the 15 evaluated baselines on both identity metrics, reducing mean CLIP reference cosine distance by 55% relative to the strongest baseline.

## 2 Related Work

**Implicit Geometry in RGB Compositing.** Paint-by-Example (Yang et al., 2023) and AnyDoor (Chen et al., 2024b) transfer reference appearance; Personalize-Anything (Feng et al., 2026), SHINE (Lu et al., 2026), and HiFi-Inpaint (Liu et al., 2026) improve identity through token substitution, sampling-time control, and high-frequency detail cues (*e.g.*, edges), respectively. Such 2D structural cues can preserve local shape but do not specify scene-wide composite depth. D2R predicts this geometric condition before RGB generation. Insert-Anything (Song et al., 2026) provides our direct baseline because its in-context canvas and LoRA (Hu et al., 2022) support a matched evaluation of adding depth prediction and supervision.

Table 1: **Geometry for Reference-Based Object Compositing.** Standard inputs are background and reference images plus a 2D compositing mask. Fixed geometry denotes depth maps or RGB proxies supplied before RGB generation.

| Method | Geometry Condition | Standard Inputs Only | Predicts Composite Depth | Fixed Geometry Before RGB |
|---|---|---|---|---|
| Insert-Anything AAAI'26 | Implicit in RGB | ✓ | ✗ | ✗ |
| Personalize-Anything AAAI'26 | Implicit in RGB | ✓ | ✗ | ✗ |
| SHINE ICLR'26 | Implicit in RGB | ✓ | ✗ | ✗ |
| HiFi-Inpaint CVPR'26 | Implicit in RGB | ✓ | ✗ | ✗ |
| MimicBrush NeurIPS'24 | Input Background Depth | ✓ | ✗ | ✓ |
| BIFRÖST NeurIPS'24 | Rule-Assembled Depth | ✗ | ✗ | ✓ |
| DIRECT ICML'26 | Proxy-Rendered RGB | ✗ | ✗ | ✓ |
| **D2R (Ours)** | Predicted Composite Depth | ✓ | ✓ | ✓ |

**Geometry Before RGB Synthesis.** MimicBrush (Chen et al., 2024a) optionally uses depth from the input background. BIFRÖST (Li et al., 2024a) assembles reference and scene depth in its instruction branch; we evaluate its mask-based inpainting branch. DIRECT (Gong et al., 2026b) renders an RGB proxy from a user-posed 3D object. These conditions describe observed geometry or a supplied arrangement. D2R instead predicts depth for the intended composite by correcting a frozen estimator's features (Tab. 1). Geometry-first approaches, such as 3DIS (Zhou et al., 2025), share the depth-before-RGB ordering; App. A distinguishes their task, inputs, and prediction objectives.

**Geometry During Sampling.** TRACE (Jo et al., 2026) finds instance structure in intermediate diffusion states, motivating our focus on when geometry becomes available. FreeInpaint (Gong et al., 2026a) optimizes noise and intermediate latents; closed-loop control (Zhao et al., 2026) adjusts conditioning using feedback from generated outputs. Both control methods adjust RGB generation at inference, whereas D2R predicts composite depth once and holds it fixed throughout RGB sampling. Section 5 compares D2R with compositing-specific optimization and joint-denoising baselines.

**Encoder Adaptation and RGB Conditioning.** ViT-Adapter (Chen et al., 2023) and ViT-CoMer (Xia et al., 2024) augment encoders for dense prediction; auxiliary branches and canvas patches condition RGB generators (Zhang et al., 2023; Mou et al., 2024; Tan et al., 2025). VGGT-World (Sun et al., 2026) predicts future geometry features from video observations and decodes them through frozen geometry components. D2R instead predicts the representation of a requested static composite from a background, reference, and compositing mask. Appendix A.3 compares these targets and the connections to joint RGB-depth modeling.

## 3 METHOD

Let $X = (I_{\text{bg}}, I_{\text{ref}}, M)$ contain the background, reference, and 2D compositing mask, with $M = 1$ on the target region. Reference cropping uses its dataset-provided object mask (see App. B.3.1). For a paired composite $I$, a frozen estimator $F$ supplies $D = F(I)$, termed *ground-truth (GT) composite depth*. This target is estimator-derived relative depth, not sensor ground truth. Stage 1 predicts $\hat{D} = \Phi_\phi(X)$ with trainable injector parameters $\phi$. D2R uses the following factorization:

$$p(I \mid X) = \int p(I \mid D, X)\, p(D \mid X)\, \mathrm{d}D, \qquad \hat{D} = \Phi_\phi(X). \tag{1}$$

D2R approximates $p(D \mid X)$ with one prediction $\hat{D}$ and generates RGB from $p_\theta(I \mid \hat{D}, X)$, where $\theta$ denotes trainable LoRA parameters (Fig. 2). Appendix B.1 defines the sampling notation.

### 3.1 STAGE 1: PREDICTING COMPOSITE DEPTH FROM REFERENCE AND BACKGROUND

Stage 1 uses a paired composite's encoder features as targets because the pretrained decoder already maps them to composite depth. We freeze the encoder $E$ and decoder $G$ of $F = G \circ E$ and train

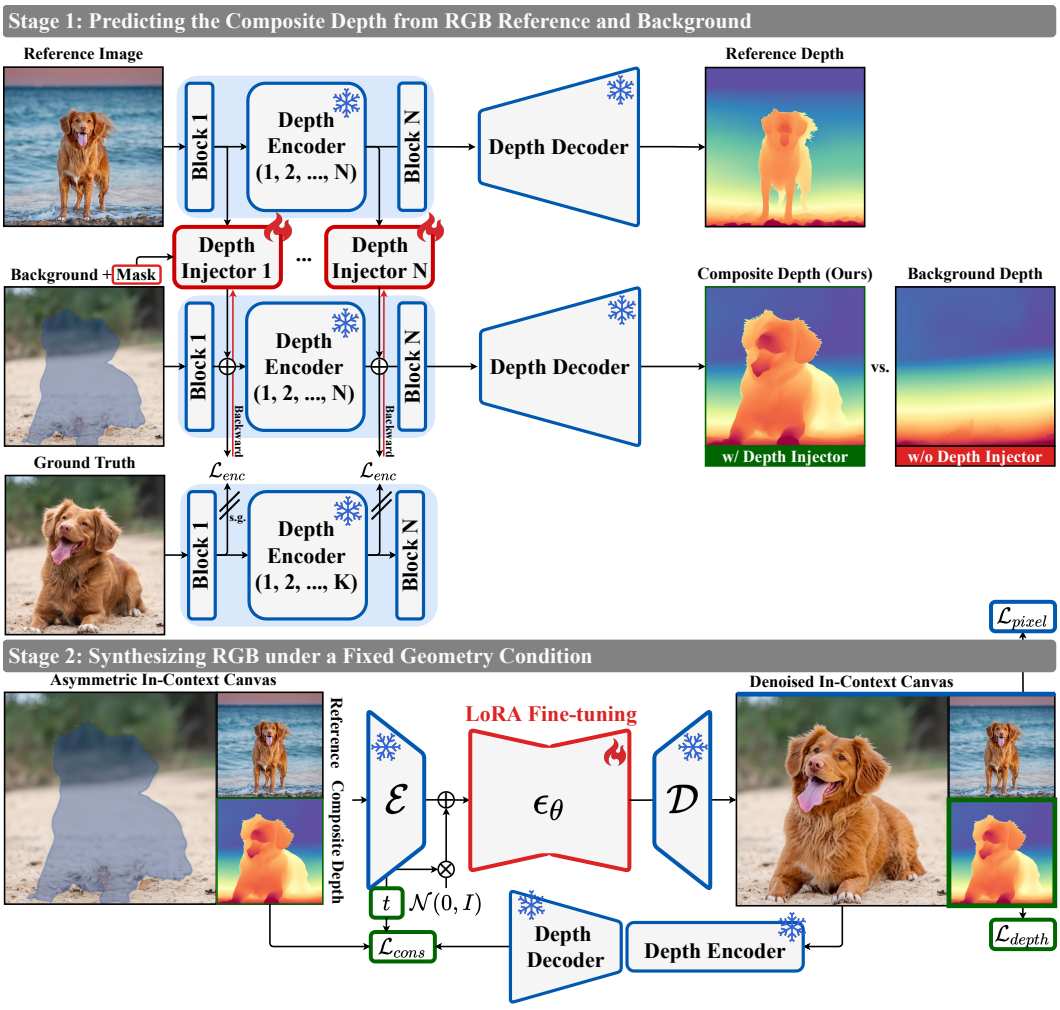


Figure 2: **Overview of Depth-to-RGB (D2R).** Stage 1 (Sec. 3.1) predicts composite depth through reference-conditioned corrections to background features in a frozen depth estimator. Stage 2 (Sec. 3.2) synthesizes RGB with this depth held fixed throughout sampling. During training, the depth input $D_{\text{in}}$ is either a Stage-1 prediction or the paired target depth $D$. RGB-depth consistency in Eq. 3 always uses $D$ as supervision, allowing the renderer to learn from imperfect depth inputs without treating their errors as geometric targets.

mask-conditioned injectors. At encoder blocks $\mathcal{K}$ read by $G$, each injector attends to background and reference features and adds a residual $\delta^\ell$:

$$\tilde{h}^\ell_{\text{bg}} = B_\ell(\tilde{h}^{\ell-1}_{\text{bg}}) + \mathbf{1}[\ell \in \mathcal{K}]\,\delta^\ell, \qquad \hat{D} = G\big(\{\tilde{h}^\ell_{\text{bg}}\}_{\ell\in\mathcal{K}}\big),$$
$$\mathcal{L}_{\text{enc}} = \frac{1}{|\mathcal{K}|}\sum_{\ell\in\mathcal{K}} \text{mean}_b \left[\frac{\text{mean}\big|\tilde{h}^\ell_{\text{bg},b} - \text{sg}(h^\ell_{\text{gt},b})\big|}{\text{sg}(\sigma^\ell_b) + \varepsilon_{\text{enc}}}\right]. \tag{2}$$

Here $B_\ell$ is a frozen encoder block and $\tilde{h}^\ell_{\text{bg}}$ denotes corrected features initialized from masked-background tokens at $\ell = 0$. The indicator selects $\mathcal{K}$, and sg stops gradients. A training-only pass through $E$ supplies targets $h^\ell_{\text{gt}}$. For sample $b$, $\sigma^\ell_b$ is their standard deviation over channels and spatial tokens; the inner error mean uses the same dimensions and $\varepsilon_{\text{enc}} = 10^{-6}$ stabilizes division. Appendix B.2.4 compares feature targets with decoded-depth supervision and their combination.

Corrections span the full token grid and propagate through subsequent frozen blocks, allowing object and background geometry to be inferred jointly. Zero-initialized output projections initially recover the frozen estimator on the same masked-background input. Each depth backbone uses the same injector design with adapted attachment points and separately trained parameters. Appendix B.2 specifies attention, gates, and architecture-specific connections; App. C.2 gives training details.

### 3.2 Stage 2: Synthesizing RGB with Fixed Composite Depth

Following canvas-based conditioning (Song et al., 2026; Tan et al., 2025), Stage 2 arranges the masked background, reference image, and composite depth as composite, reference, and depth patches on one input canvas. At inference, RGB is generated only in the masked region of the composite patch; the rest of the canvas remains fixed as conditioning input (App. B.3.1).

**Training and Auxiliary Supervision.** Auxiliary depth prediction trains the shared denoiser to represent composite geometry. A frozen autoencoder encodes the input canvas and a supervision canvas whose three patches contain the paired composite, reference image, and target depth $D$; only denoiser LoRA (Hu et al., 2022) is trained. The input depth patch receives $\hat{D}$ or $D$ with equal probability, exposing the renderer to prediction errors while retaining the same supervision. We add noise to the full supervision canvas during training and minimize $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pixel}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{cons}}$. The RGB loss supervises the composite and reference patches; the depth loss supervises the depth patch. At inference, only the masked region of the composite patch is generated, while the supplied depth patch remains fixed. An accurate depth patch does not ensure that the generated RGB image matches the target geometry. We therefore apply the frozen estimator $F$ to the one-step clean RGB estimate $\hat{I}_{\text{comp}}$ and compare its depth with the paired target $D$:

$$\mathcal{L}_{\text{cons}} = (1-t)\left\| A(F(\hat{I}_{\text{comp}})) - \text{sg}(A(D)) \right\|_1. \tag{3}$$

$A$ subtracts each map's median and divides by its mean absolute deviation from that median, using statistics over the entire composite patch. The loss averages over valid pixels and always targets $D$, including when the input is $\hat{D}$. With $t = 1$ denoting noise and $t = 0$ clean data, $1 - t$ downweights noisy early estimates. Gradients flow through the frozen RGB decoder and depth estimator $F$ to update only the LoRA parameters. Appendix B.3 defines depth normalization and RGB decoding; App. B.1 gives one-step clean-latent estimates for rectified flow and variance-preserving diffusion.

**Inference.** Stage 1 predicts $\hat{D}$ once, then Stage 2 synthesizes the masked RGB region while restoring the depth condition after every update. The default pipeline takes 22.40 s per image on RTX PRO 6000 and preserves the background image's resolution (see App. B.4 for details).

## 4 Experiments

**Datasets.** AnyInsertion++ contains 52.9K curated training samples from AnyInsertion (Song et al., 2026) and 426 paired evaluation cases: 287 in-distribution (ID) and 139 from categories absent from compositing training (OOD). Thumbnail-hash and confirmed perceptual-hash checks find no train/test image matches. For reference-identity evaluation without paired targets, TF-ICON (Lu et al., 2023), AIComposer (Li et al., 2025), and ComplexCompo (Lu et al., 2026) provide 332, 272, and 300 cases. Target regions use object masks in AnyInsertion++ and coarse box masks in the unpaired benchmarks. Appendix C.1 details curation, overlap checks, and evaluation protocols.

**Implementation Details.** Stage 1 uses a width-640 injector with frozen Depth Anything V2 ViT-L (Yang et al., 2024b). Stage 2 adapts FLUX.1-Fill with Redux[1], following the LoRA and sampling setup of Insert-Anything (Song et al., 2026). All trained variants use final checkpoints after 50K iterations for Stage 1 and 100K for Stage 2. Insert-Anything* is retrained on the same AnyInsertion++ data. Other baselines use released checkpoints or public interfaces. Settings and seed checks are in App. C.3.2; paired uncertainty and scorer robustness are in App. C.4.

**Metrics.** Full-frame AbsRel, $\delta_1$, and RMSE compare depths estimated from GT and generated RGB after background-based scale and shift alignment. PSNR and LPIPS (Zhang et al., 2018b) measure photometric quality against GT composites. CLIP-I and DINO-I use IsoCLIP (Magistri et al., 2026) and DINOv3 ViT-L (Siméoni et al., 2025), respectively, to measure cosine similarity between generated object crops and GT crops (paired) or reference crops (unpaired; App. C.4.4).

**Paired and Unpaired Performance.** D2R outperforms all 15 open- and closed-source baselines on every reported geometric, photometric, and identity metric across both paired splits (Tabs. 2 and 3). Compared with the strongest baseline for each metric, it reduces AbsRel by at least 22% and LPIPS by at least 19%, while improving PSNR by at least 1.0 dB. D2R also leads both identity

[1]Official model cards: FLUX.1-Fill and FLUX.1-Redux.

Table 2: In-Distribution Results on AnyInsertion++.

| Method | Backbone | Geometric | | | Photometric | | Identity | |
|---|---|---|---|---|---|---|---|---|
| | | AbsRel↓ | $\delta_1$ ↑ | RMSE↓ | PSNR↑ | LPIPS↓ | CLIP-I↑ | DINO-I↑ |
| Paint-by-Example CVPR'23 | SD1.5 | 0.112 | 0.885 | 0.085 | 17.7 | 0.305 | 0.682 | 0.427 |
| TF-ICON ICCV'23 | SD2.1 | 0.205 | 0.779 | 0.118 | 17.2 | 0.414 | 0.545 | 0.262 |
| AnyDoor CVPR'24 | SD2.1 | 0.173 | 0.841 | 0.091 | 17.5 | 0.370 | 0.770 | 0.634 |
| MimicBrush NeurIPS'24 | SD1.5 | 0.066 | 0.950 | 0.048 | 19.5 | 0.218 | 0.771 | 0.623 |
| BIFRÖST NeurIPS'24 | SD2.1 | 0.099 | 0.635 | 0.056 | 19.4 | 0.314 | 0.764 | 0.649 |
| DreamFuse ICCV'25 | FLUX.1 | 0.127 | 0.887 | 0.077 | 17.8 | 0.334 | 0.738 | 0.612 |
| UniCombine ICCV'25 | FLUX.1 | 0.073 | 0.950 | 0.046 | 17.6 | 0.289 | 0.754 | 0.654 |
| Personalize-Anything AAAI'26 | FLUX.1 | 0.177 | 0.835 | 0.104 | 17.5 | 0.459 | 0.706 | 0.554 |
| Insert-Anything* AAAI'26 | FLUX.1 | 0.062 | 0.963 | 0.036 | 21.2 | 0.213 | 0.887 | 0.909 |
| SHINE ICLR'26 | FLUX.1 | 0.162 | 0.840 | 0.092 | 17.2 | 0.331 | 0.652 | 0.493 |
| HiFi-Inpaint CVPR'26 | FLUX.1 | 0.104 | 0.905 | 0.076 | 19.0 | 0.262 | 0.757 | 0.661 |
| DIRECT ICML'26 | FLUX.1 | 0.088 | 0.922 | 0.069 | 19.4 | 0.249 | 0.832 | 0.759 |
| GPT-Image-1 Tested 2026-08 | Proprietary | 0.468 | 0.568 | 0.162 | 12.6 | 0.607 | 0.765 | 0.767 |
| GPT-Image-2 Tested 2026-08 | Proprietary | 0.206 | 0.796 | 0.113 | 15.0 | 0.432 | 0.872 | 0.888 |
| Nano Banana 2 Tested 2026-08 | Proprietary | 0.175 | 0.844 | 0.094 | 16.1 | 0.365 | 0.901 | 0.874 |
| **D2R (Predicted Depth)** | FLUX.1 | **0.048** | **0.978** | **0.028** | **22.2** | **0.172** | **0.927** | **0.931** |
| D2R (Ground-Truth Depth) | FLUX.1 | 0.044 | 0.982 | 0.023 | 22.7 | 0.167 | 0.931 | 0.938 |

Table 3: Category-Disjoint Results on AnyInsertion++.

| Method | Backbone | Geometric | | | Photometric | | Identity | |
|---|---|---|---|---|---|---|---|---|
| | | AbsRel↓ | $\delta_1$ ↑ | RMSE↓ | PSNR↑ | LPIPS↓ | CLIP-I↑ | DINO-I↑ |
| Paint-by-Example CVPR'23 | SD1.5 | 0.136 | 0.871 | 0.074 | 17.6 | 0.258 | 0.676 | 0.401 |
| TF-ICON ICCV'23 | SD2.1 | 0.382 | 0.633 | 0.135 | 16.9 | 0.366 | 0.542 | 0.217 |
| AnyDoor CVPR'24 | SD2.1 | 0.258 | 0.752 | 0.096 | 17.2 | 0.308 | 0.714 | 0.503 |
| MimicBrush NeurIPS'24 | SD1.5 | 0.090 | 0.926 | 0.049 | 20.2 | 0.174 | 0.795 | 0.669 |
| BIFRÖST NeurIPS'24 | SD2.1 | 0.120 | 0.905 | 0.052 | 19.4 | 0.243 | 0.731 | 0.565 |
| DreamFuse ICCV'25 | FLUX.1 | 0.227 | 0.799 | 0.089 | 17.3 | 0.301 | 0.746 | 0.675 |
| UniCombine ICCV'25 | FLUX.1 | 0.190 | 0.839 | 0.079 | 19.3 | 0.226 | 0.796 | 0.764 |
| Personalize-Anything AAAI'26 | FLUX.1 | 0.188 | 0.830 | 0.074 | 19.7 | 0.322 | 0.704 | 0.602 |
| Insert-Anything* AAAI'26 | FLUX.1 | 0.119 | 0.914 | 0.048 | 21.6 | 0.176 | 0.863 | 0.887 |
| SHINE ICLR'26 | FLUX.1 | 0.218 | 0.827 | 0.088 | 17.5 | 0.295 | 0.647 | 0.443 |
| HiFi-Inpaint CVPR'26 | FLUX.1 | 0.133 | 0.893 | 0.067 | 18.7 | 0.235 | 0.746 | 0.665 |
| DIRECT ICML'26 | FLUX.1 | 0.137 | 0.877 | 0.073 | 19.4 | 0.221 | 0.828 | 0.749 |
| GPT-Image-1 Tested 2026-08 | Proprietary | 0.492 | 0.591 | 0.149 | 13.2 | 0.511 | 0.786 | 0.782 |
| GPT-Image-2 Tested 2026-08 | Proprietary | 0.300 | 0.735 | 0.116 | 14.8 | 0.407 | 0.902 | 0.912 |
| Nano Banana 2 Tested 2026-08 | Proprietary | 0.237 | 0.812 | 0.091 | 16.5 | 0.328 | 0.922 | 0.905 |
| **D2R (Predicted Depth)** | FLUX.1 | **0.067** | **0.970** | **0.030** | **24.0** | **0.117** | **0.923** | **0.916** |
| D2R (Ground-Truth Depth) | FLUX.1 | 0.059 | 0.977 | 0.024 | 24.8 | 0.108 | 0.927 | 0.929 |

metrics on all three unpaired benchmarks, reducing mean reference cosine distance by 55.4% with CLIP and 22.6% with DINO over Insert-Anything* (Tab. 4; App. C.1.3). These gains occur *without an identity-specific loss*. Objects with similar semantics and colors can differ in shape and part arrangement. Our qualitative examples show that D2R better preserves these structural cues; we associate this preservation with the higher reference embedding similarity, as shown in Fig. 1.

**Geometry from Standard Compositing Inputs.** On the paired splits, replacing Stage-1 predictions with GT composite depth lowers rendered-image AbsRel by about 8 to 12% and raises PSNR by 0.5 to 0.8 dB. The predicted condition already supports leading performance on unseen categories without a completed target image, measured depth, or a user-posed 3D proxy at inference.

Unpaired benchmarks further assess reference identity across real-world photographs and stylized scenes, extending evaluation beyond paired image comparisons. Surprisingly, the same pipeline also produces geometry-aware composites in cartoon and artistic scenes. A physical depth sensor viewing a flat picture or screen measures that surface, rather than distances within the depicted scene. A learned depth estimator instead infers relative depth from RGB cues. D2R repurposes

Table 4: Identity on Unpaired Compositing Benchmarks.

| Method | Backbone | TF-ICON | | AIComposer | | ComplexCompo | |
|---|---|---|---|---|---|---|---|
| | | CLIP-I↑ | DINO-I↑ | CLIP-I↑ | DINO-I↑ | CLIP-I↑ | DINO-I↑ |
| Paint-by-Example CVPR'23 | SD1.5 | 0.689 | 0.342 | 0.696 | 0.387 | 0.528 | 0.223 |
| TF-ICON ICCV'23 | SD2.1 | 0.573 | 0.258 | 0.654 | 0.374 | 0.489 | 0.199 |
| AnyDoor CVPR'24 | SD2.1 | 0.752 | 0.628 | 0.777 | 0.625 | 0.718 | 0.645 |
| MimicBrush NeurIPS'24 | SD1.5 | 0.753 | 0.537 | 0.767 | 0.575 | 0.645 | 0.415 |
| BIFRÖST NeurIPS'24 | SD2.1 | 0.730 | 0.557 | 0.736 | 0.548 | 0.685 | 0.565 |
| DreamFuse ICCV'25 | FLUX.1 | 0.652 | 0.410 | 0.718 | 0.597 | 0.650 | 0.569 |
| UniCombine ICCV'25 | FLUX.1 | 0.697 | 0.555 | 0.747 | 0.641 | 0.482 | 0.297 |
| Personalize-Anything AAAI'26 | FLUX.1 | 0.739 | 0.542 | 0.808 | 0.716 | 0.668 | 0.615 |
| Insert-Anything* AAAI'26 | FLUX.1 | 0.760 | 0.679 | 0.816 | 0.794 | 0.778 | 0.724 |
| SHINE ICLR'26 | FLUX.1 | 0.402 | 0.034 | 0.469 | 0.155 | 0.557 | 0.376 |
| HiFi-Inpaint CVPR'26 | FLUX.1 | 0.468 | 0.116 | 0.583 | 0.352 | 0.449 | 0.219 |
| DIRECT ICML'26 | FLUX.1 | 0.714 | 0.529 | 0.742 | 0.595 | 0.742 | 0.683 |
| GPT-Image-1 Tested 2026-08 | Proprietary | 0.612 | 0.416 | 0.645 | 0.567 | 0.469 | 0.241 |
| GPT-Image-2 Tested 2026-08 | Proprietary | 0.625 | 0.450 | 0.669 | 0.608 | 0.456 | 0.178 |
| Nano Banana 2 Tested 2026-08 | Proprietary | 0.591 | 0.345 | 0.632 | 0.487 | 0.413 | 0.150 |
| **D2R (Predicted Depth)** | FLUX.1 | **0.902** | **0.783** | **0.915** | **0.813** | **0.895** | **0.781** |

this capability to predict the intended composite's geometry and guide RGB synthesis. Appendix D discusses this observation through the cartoon and illustrated examples in Figs. D3 and D4.

## 5 Ablation Study and Discussion

Table 5: Effect of the Supervision Target with a Matched Depth Predictor.

| Supervision Target | Decoder Backpropagation | AbsRel ↓ | | $\delta_1$ ↑ | |
|---|---|---|---|---|---|
| | | ID | OOD | ID | OOD |
| **Encoder Feature Only** | ✗ | **0.0462** | **0.0751** | **0.977** | **0.965** |
| Decoded Depth Only | ✓ | 0.0669 | 0.1094 | 0.955 | 0.931 |

**Learning the Completed Scene's Depth Features.** Table 5 compares encoder-feature and decoded-depth supervision with the same injector, frozen estimator, data, and training settings. Both the encoder and decoder remain frozen; only the injector is trained. Encoder-feature supervision directly guides background features toward those of the paired composite. Decoded-depth supervision constrains the output map through decoder backpropagation, but different feature configurations can yield similar depth. Before RGB synthesis, feature supervision lowers ID and OOD Stage-1 AbsRel by 30.9% and 31.4%, respectively, while avoiding that backward pass. Adding decoded-depth loss to feature matching lowers ID error by about 5% but raises OOD error by about 3% and reduces the largest single-GPU batch by about 30%. The feature-supervised predictor also lowers rendered-image OOD AbsRel from 0.089 to 0.067 with the same Stage-2 RGB renderer (App. B.2.4).

**Feature Correction Versus Diffusion-Based Depth Generation.** Following the depth-generation approach in 3DIS (Zhou et al., 2025), we build a compositing-specific diffusion baseline with FLUX.1-Fill and Redux. It denoises composite-depth maps using the same Stage-1 training pairs. D2R (Sec. 3.1) reduces OOD AbsRel from 0.1740 to 0.0751 (56.8%) with one feed-forward pass instead of 28 denoising steps. This compares depth-prediction strategies; Tab. 5 isolates supervision within the same predictor (App. C.2.2).

**Predicting Geometry Before RGB Synthesis.** The matched RGB-only baseline in Tab. 6(a) shares D2R's data and setup but infers object structure during appearance generation, reaching OOD AbsRel of 0.119 and PSNR of 21.6 dB. Table 6(b) lowers AbsRel to 0.098 but reduces PSNR to 20.3 dB. Rescaling reference depth into the background supplies observed geometry without predicting the object's shape in its new scene, so this cue can conflict with the intended composite.

Latent optimization in Tab. 6(c) already targets Stage-1 depth, yet worsens both scores to 0.224 and 16.9 dB. It adjusts latents using depth estimated from provisional RGB images without training the renderer to follow that condition. Such depth-driven updates can be unreliable before object

Table 6: Geometric and Photometric Performance Across Geometry Strategies.

| Geometry Provisioning | Out-of-Distribution | | In-Distribution | |
|---|---|---|---|---|
| | AbsRel ↓ | PSNR ↑ | AbsRel ↓ | PSNR ↑ |
| (a) Empty Depth Patch (Matched RGB Baseline) | 0.119 | 21.6 | 0.062 | 21.2 |
| (b) Heuristic Depth Fusion | 0.098 | 20.3 | 0.076 | 18.9 |
| (c) Latent Optimization Toward Predicted Depth | 0.224 | 16.9 | 0.139 | 17.3 |
| (d) Joint Denoising from Background Depth | 0.080 | 22.2 | 0.065 | 20.9 |
| **(e) Sequential Synthesis with Fixed Depth (D2R; Ours)** | **0.067** | **24.0** | **0.048** | **22.2** |

Table 7: Effect of Depth Conditioning and Auxiliary Supervision.

| | $\mathcal{L}_{\text{pixel}}$ | $\mathcal{L}_{\text{depth}}$ | $\mathcal{L}_{\text{cons}}$ | Geometric | | | Photometric | |
|---|---|---|---|---|---|---|---|---|
| | | | | AbsRel ↓ | $\delta_1$ ↑ | RMSE ↓ | PSNR ↑ | LPIPS ↓ |
| (a) | ✓ | ✗ (w/o depth input) | ✗ | 0.119 | 0.914 | 0.048 | 21.6 | 0.176 |
| (b) | ✓ | ✗ (w/ depth input) | ✗ | 0.108 | 0.931 | 0.046 | 21.9 | 0.168 |
| (c) | ✓ | ✓ | ✗ | 0.081 | 0.958 | 0.038 | 23.1 | 0.127 |
| (d) | ✓ | ✗ | ✓ $(1-t)$ | 0.084 | 0.947 | 0.043 | 22.1 | 0.158 |
| (e) | ✓ | ✓ | ✓ (uniform) | 0.075 | 0.950 | 0.041 | 22.8 | 0.135 |
| **(f)** | ✓ | ✓ | ✓ $(1-t)$ | **0.067** | **0.970** | **0.030** | **24.0** | **0.117** |

structure emerges and do not explicitly enforce reference appearance. Joint denoising in Tab. 6(d) reaches OOD AbsRel of 0.080 but raises ID error from 0.062 to 0.065: starting from background depth leaves composite geometry to be resolved alongside RGB. In Tab. 6(e), D2R instead predicts that geometry first and trains RGB synthesis under a fixed condition, improving both metrics on both splits. These results compare complete implementations: geometry construction and training also differ, so they do not isolate generation order (App. C.3).

**Learning to Use the Depth Condition.** Table 7 first tests depth input under RGB-only supervision: OOD AbsRel falls from 0.119 to 0.108 (a to b). With depth supply fixed, depth-patch denoising and RGB-depth consistency reduce error further to 0.081 and 0.084, respectively (b to c/d). The former supervises the depth patch; the latter compares geometry estimated from generated RGB with the paired target. Combining them improves on either objective alone (c/d to f). With both enabled, replacing uniform consistency weighting with $1 - t$ lowers AbsRel from 0.075 to 0.067 and raises PSNR from 22.8 to 24.0 dB (e to f), reducing the influence of noisy early RGB estimates. All conditioned variants use Stage-1 predictions at evaluation (App. C.3.4).

**Learning from Accurate and Predicted Depth.** Stage 2 must handle errors in the depth supplied at inference. Table 8 varies training inputs with GT supervision fixed and scores generated RGB geometry. With GT-only training, OOD AbsRel rises from 0.062 under GT depth to 0.081 under predictions. Predicted-only training improves the latter to 0.069 but weakens performance with GT depth. Blending both sources can average mismatched object boundaries. Selection instead pairs either intact map with the same GT target. This exposes the renderer to prediction errors without using them as supervision targets. Selection gives the lowest error under both input types: OOD AbsRel of 0.059 with GT depth and 0.067 with predictions (App. B.3).

**Correcting Object and Scene Features Together.** The mask localizes the compositing region, but relative depth also depends on its surroundings. Restricting correction to masked tokens raises average Stage-1 AbsRel by about 2.5%, with errors increasing inside and outside the mask. Sharing one injector across four layers raises ID and OOD error by about 10% and 3%. These results favor scene-wide correction with separate layer parameters, letting reference information influence both object and context features (App. B.2). Injection also improves all six depth backbones against their own targets, supporting reuse across estimators without ranking absolute accuracy (Tab. C4).

**Adapting Different RGB Generators.** We supply identical Stage-1 depth to pretrained and adapted SD2 (Rombach et al., 2022), SDXL (Podell et al., 2024), Qwen (Wu et al., 2025), and FLUX.1-Fill. Adapted configurations reduce OOD AbsRel by 18% to 32% and increase PSNR by 1.08 to 2.13 dB (Tab. 9). Because both versions receive the same depth, the gains reflect improved RGB generation under that condition. This supports reuse of the depth interface across four generators (App. C.3.3).

Table 8: Training the Renderer to Handle Depth Prediction Errors.

| Depth Supply Strategy | Ground-Truth Depth at Inference | | Predicted Depth at Inference | |
|---|---|---|---|---|
| | ID AbsRel ↓ | OOD AbsRel ↓ | ID AbsRel ↓ | OOD AbsRel ↓ |
| Ground-Truth Depth Only | 0.049 | 0.062 | 0.056 | 0.081 |
| Predicted Depth Only | 0.052 | 0.066 | 0.053 | 0.069 |
| Blending (*i.e.*, mixup (Zhang et al., 2018a)) | 0.051 | 0.064 | 0.055 | 0.070 |
| Selection, $p = 0.5$ | **0.044** | **0.059** | **0.048** | **0.067** |

Table 9: Depth-Conditioned RGB Synthesis Across Generators.

| Diffusion Backbone | AbsRel ↓ | | | $\delta_1$ ↑ | | PSNR (dB) ↑ | | |
|---|---|---|---|---|---|---|---|---|
| | Pretrained | + D2R | Δ | Pretrained | + D2R | Pretrained | + D2R | Δ |
| SD2-Inpainting | 0.158 | **0.110** | −31% | 0.830 | **0.925** | 18.10 | **19.57** | +1.47 |
| SDXL-Inpainting | 0.155 | **0.105** | −32% | 0.832 | **0.930** | 18.00 | **20.13** | +2.13 |
| Qwen-Inpainting | 0.100 | **0.082** | −18% | 0.920 | **0.954** | 20.47 | **21.55** | +1.08 |
| FLUX.1-Fill | 0.088 | **0.067** | −23% | 0.942 | **0.970** | 21.97 | **24.02** | +2.05 |

**Consistency Across Cases and Object Regions.** D2R improves OOD full-frame AbsRel over Insert-Anything* on 103 of 139 cases, with gains inside objects and along their boundaries. Appendix C.4.2 reports paired and image-cluster uncertainty and sensitivity to the largest gains. Five default training runs per stage show about 1% average relative variation across geometric, photometric, and identity metrics (App. C.3.2).

**Generalization Beyond Rigid Objects.** To test whether gains extend to objects whose shape can change, we evaluate the annotated rigid and non-rigid groups separately. D2R improves OOD AbsRel and PSNR over Insert-Anything* in both groups; non-rigid AbsRel falls from 0.133 to 0.071. This evidence covers 20 non-rigid OOD cases (App. C.1, Tab. C3).

**Agreement Across Depth Scorers.** We hold generated RGB fixed and change only the depth scorer. D2R ranks first among sixteen methods on both splits under all five scorers, including Depth Pro and Metric-Anything, which supply no training targets to the default model (App. C.4).

**Computational Cost.** On RTX PRO 6000, the complete Stage-1 predictor takes 56.5 ms and 5.21 GB peak inference memory. Relative to 22.34 s for FLUX.1-Fill RGB synthesis, this adds about 0.25% latency. Predicted composite depth is reused throughout all 28 RGB sampling steps. The injector adds 272.4M parameters. The lower-resolution SD2 configuration takes 0.87 s overall, trading geometric and photometric quality for speed (see Tab. 9; App. B.4 for details).

**Limitations.** D2R keeps its predicted depth fixed during RGB synthesis, so errors in object shape or placement can persist in the final image. Even with plausible geometry, an object may retain reference colors or textures that do not match a grayscale or illustrated background. Depth describes spatial structure but does not specify the color and texture changes needed to match the background. Appendix C.5 provides examples and further discussion.

## 6 Conclusion

D2R predicts composite depth by learning reference-conditioned corrections toward the encoder features of a completed scene while keeping the depth estimator frozen. Matched comparisons connect feature supervision to better composite depth and final RGB geometry under the same renderer. The complete pipeline improves photometric quality and reference identity. The depth predictor also outperforms our diffusion-based composite-depth baseline without iterative denoising. Feature correction adapts multiple depth estimators, while the resulting depth condition supports different RGB generators. These results extend depth estimation from observed images to intended composites, providing a reusable geometric condition for RGB synthesis without a depth sensor or user-posed 3D proxy. We believe this interface can support future harmonization and relighting that adapt object color and illumination to the surrounding scene while preserving the intended object shape and its spatial relationships with nearby surfaces.

## AI USE STATEMENT

Large language models (LLMs) were used to edit author-written text for grammar, concision, readability, and academic tone, and to assist with literature retrieval and discovery by suggesting search terms and potentially relevant prior work. The authors independently located and read the original sources and verified all citations and literature-based statements. In addition, Claude Opus 5 was used during dataset curation to score annotation validity for each candidate training sample; the authors set the filtering threshold by manually inspecting samples across score tiers and take responsibility for the final filtering decisions (App. C.1.1). LLMs were not otherwise used to draft manuscript sections or to contribute to method development, experimental design, implementation, or interpretation of results. All AI-assisted work was reviewed by the authors, who take full responsibility for the final content of the paper.

## REPRODUCIBILITY STATEMENT

We provide the complete D2R codebase, including training, inference, and evaluation pipelines, together with configuration files, environment specifications, and execution instructions in the accompanying supplementary material. Specifically, the submission includes the modified AnyInsertion++ training dataset structure alongside designated directories for paired and unpaired performance evaluations. The appendix documents the model architecture and objectives, benchmark construction and filtering, category-disjoint split protocol, comparator settings, evaluation metrics, and statistical tests. For external comparisons, we identify which methods are run from official public checkpoints, which baseline is retrained under matched settings, and which proprietary systems are evaluated through their available interfaces. Together, these materials support independent reproduction of D2R and auditing of the reported comparisons.

## REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth Pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiaxuan Chen, Bo Zhang, Qingdong He, Jinlong Peng, and Li Niu. CareCom: Generative image composition with calibrated reference features. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 2877–2885, 2026.

Xi Chen, Yutong Feng, Mengting Chen, Yiyang Wang, Shilong Zhang, Yu Liu, Yujun Shen, and Hengshuang Zhao. Zero-shot image editing with reference imitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. AnyDoor: Zero-shot object-level image customization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *International Conference on Learning Representations (ICLR)*, 2023.

Jan-Niklas Dihlmann, Andreas Engelhardt, and Hendrik Lensch. SIGNeRF: Scene integrated generation for neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6679–6688, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Dihlmann_SIGNeRF_Scene_Integrated_Generation_for_Neural_Radiance_Fields_CVPR_2024_paper.html.

Bardienus Pieter Duisterhof, Deva Ramanan, Jeffrey Ichnowski, Justin Johnson, and Keunhong Park. Modality forcing for scalable spatial generation. *arXiv preprint arXiv:2606.13676*, 2026. doi: 10.48550/arXiv.2606.13676. URL https://arxiv.org/abs/2606.13676.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 12606–12633, 2024.

Haoran Feng, Zehuan Huang, Lin Li, and Lu Sheng. Personalize anything for free with diffusion transformer. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 3921–3929, 2026.

Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Xingqian Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. PAIR Diffusion: A comprehensive multimodal object-level image editor. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8609–8618, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Goel_PAIR_Diffusion_A_Comprehensive_Multimodal_Object-Level_Image_Editor_CVPR_2024_paper.html.

Chao Gong, Dong Li, Yingwei Pan, Jingjing Chen, Ting Yao, and Tao Mei. FreeInpaint: Tuning-free prompt alignment and visual rationality enhancement in image inpainting. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 4239–4247, 2026a.

Jingbo Gong, Yikai Wang, Yushi Lan, Yuhao Wan, Ziheng Ouyang, Rui Zhao, Ming-Ming Cheng, Qibin Hou, and Chen Change Loy. Direct 3D-aware object insertion via decomposed visual proxies. In *International Conference on Machine Learning (ICML)*, 2026b.

Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/dea9b4b6f55ae611c54065d6fc750755-Abstract-Conference.html.

Yi He, Jiangming Wang, Xinyu Wang, Mark Fong, Songchun Zhang, Yuxuan Xue, Hai-Tao Zheng, and Yue Ma. GeoEdit: Geometry-aware object editing via dual-branch denoising. In *European Conference on Computer Vision (ECCV)*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Junjia Huang, Pengxiang Yan, Jiyang Liu, Jie Wu, Zhao Wang, Yitong Wang, Liang Lin, and Guanbin Li. DreamFuse: Adaptive image fusion with diffusion transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Youze Huang, Penghui Ruan, Bojia Zi, Xianbiao Qi, Jianan Wang, and Rong Xiao. Refacade: Editing object with given reference texture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1961–1972, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Huang_Refacade_Editing_Object_with_Given_Reference_Texture_CVPR_2026_paper.html.

Sanghyun Jo, Ziseok Lee, Wooyeol Lee, Jonghyun Choi, Jaesik Park, and Kyungsu Kim. TRACE: Your diffusion model is secretly an instance edge detector. In *International Conference on Learning Representations (ICLR)*, 2026.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. BrushNet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision (ECCV)*, pp. 150–168, 2024.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9492–9502, 2024.

Byung-Ki Kwon, Qi Dai, Hyoseok Lee, Chong Luo, and Tae-Hyun Oh. JointDiT: Enhancing RGB-Depth joint modeling with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25261–25271, October 2025.

Jonghyun Lee, Hansam Cho, Youngjoon Yoo, Seoung Bum Kim, and Yonghyun Jeong. Compose and conquer: Diffusion-based 3D depth aware composable image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Guangzhao Li, Qingyan Wei, Huayu Zheng, Yige Zheng, Chaoyang Zhang, Jie Yang, Yunan Ding, Yan Tai, Siqi Luo, and Xiaohong Liu. InsertFuse: A unified framework for multi-category reference-guided image insertion. *arXiv preprint arXiv:2608.06490*, 2026.

Haowen Li, Zhenfeng Fan, Zhang Wen, Zhengzhou Zhu, and Yunjin Li. AIComposer: Any style and content image composition via feature integration. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16840–16850, 2025.

Lingxiao Li, Kaixiong Gong, Weihong Li, Xili Dai, Tao Chen, Xiaojun Yuan, and Xiangyu Yue. Bifröst: 3D-aware image compositing with language instructions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. ControlNet++: Improving conditional controls with efficient consistency feedback. In *European Conference on Computer Vision (ECCV)*, 2024b.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Yichen Liu, Donghao Zhou, Jie Wang, Xin Gao, Guisheng Liu, Jiatong Li, Quanwei Zhang, Qiang Lyu, Lanqing Guo, Shilei Wen, Weiqiang Wang, and Pheng-Ann Heng. HiFi-Inpaint: Towards high-fidelity reference-based inpainting for generating detail-preserving human-product images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. TF-ICON: Diffusion-based training-free cross-domain image composition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2294–2305, 2023.

Shilin Lu, Zhuming Lian, Zihan Zhou, Shaocong Zhang, Chen Zhao, and Adams Wai-Kin Kong. Does FLUX already know how to perform physically plausible image composition? In *International Conference on Learning Representations (ICLR)*, 2026.

Simone Magistri, Dipam Goswami, Marco Mistretta, Bartłomiej Twardowski, Joost van de Weijer, and Andrew D. Bagdanov. IsoCLIP: Decomposing CLIP projectors for efficient intra-modal alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29315–29324, 2026.

Shun Masuda, Yuki Endo, and Yoshihiro Kanamori. Person-In-Situ: Scene-consistent human image insertion with occlusion-aware pose control. *Computer Animation and Virtual Worlds*, 37(3): e70119, 2026. doi: 10.1002/cav.70119.

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35779–35804. PMLR, 2024. URL https://proceedings.mlr.press/v235/mishchenko24a.html.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 2024.

Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. Diffusion Handles enabling 3D edits for diffusion models by lifting activations to 3D. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7695–7704, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Pandey_Diffusion_Handles_Enabling_3D_Edits_for_Diffusion_Models_by_Lifting_CVPR_2024_paper.html.

Rishubh Parihar, Sachidanand VS, and R. Venkatesh Babu. Zero-shot depth aware image editing with diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15748–15759, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12179–12188, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Wensong Song, Hong Jiang, Zongxin Yang, Zheqiao Cheng, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in DiT. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 9097–9105, 2026.

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. ObjectStitch: Object compositing with diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18310–18319, 2023.

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. IMPRINT: Generative object compositing by learning identity-preserving representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8048–8058, 2024.

Xiangyu Sun, Shijie Wang, Fengyi Zhang, Lin Liu, Caiyan Jia, Ziying Song, Zi Huang, and Yadan Luo. VGGT-World: Transforming VGGT into an autoregressive geometry world model. In *European Conference on Computer Vision (ECCV)*, 2026.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and universal control for diffusion transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14940–14950, 2025.

Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Tumanyan_Plug-and-Play_Diffusion_Features_for_Text-Driven_Image-to-Image_Translation_CVPR_2023_paper.html.

Haoxuan Wang, Jinlong Peng, Qingdong He, Hao Yang, Ying Jin, Jiafu Wu, Xiaobin Hu, Yanjie Pan, Zhenye Gan, Mingmin Chi, Bo Peng, and Yabiao Wang. UniCombine: Unified multi-conditional combination with diffusion transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18325–18334, 2025.

Zehan Wang, Jialei Wang, Siyu Chen, Ziang Zhang, Luping Liu, Xize Cheng, Kaihang Pan, Hengshuang Zhao, and Zhou Zhao. SpatialHand: Generative object manipulation from 3D perspective. In *International Conference on Learning Representations (ICLR)*, 2026.

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision (ECCV)*, 2024.

Daniel Winter, Asaf Shul, Matan Cohen, Dana Berman, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectMate: A recurrence prior for object insertion and subject-driven generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. ViT-CoMer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by Example: Exemplar-based image editing with diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18381–18391, 2023.

Jiahui Yang, Donglin Di, Xuancheng Zhang, Lei Fan, Jianxun Cui, Hao Li, and Baorui Ma. MetricAnything: Scaling metric depth pretraining with noisy heterogeneous sources. In *European Conference on Computer Vision (ECCV)*, 2026.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10371–10381, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 21875–21911, 2024b.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. InfiniDepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26920–26930, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018b.

Shuo Zhang, Wenzhuo Wu, Huayu Zhang, Jiarong Cheng, Xianghao Zang, Chao Ban, Hao Sun, Zhongjiang He, Tianwei Cao, Kongming Liang, and Zhanyu Ma. Geometric image editing via effects-sensitive in-context inpainting with diffusion transformers. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://iclr.cc/virtual/2026/poster/10011948.

Yuhong Zhang, Han Wang, Yiwen Wang, Rong Xie, and Li Song. FreeInsert: Personalized object insertion with geometric and style control. In *ACM International Conference on Multimedia (ACM MM)*, 2025. doi: 10.1145/3746027.3755603.

Baixuan Zhao, Xinyu Zhang, Huayu Zheng, Shuaicheng Liu, Xiongkuo Min, Guangtao Zhai, and Xiaohong Liu. From open loop to closed loop: A test-time iterative optimization framework for reference-consistent image generation. In *European Conference on Computer Vision (ECCV)*, 2026.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Huayu Zheng, Guangzhao Li, Baixuan Zhao, Siqi Luo, Hantao Jiang, Guangtao Zhai, and Xiaohong Liu. $A^2$-Edit: Precise reference-guided image editing of arbitrary objects and ambiguous masks. In *European Conference on Computer Vision (ECCV)*, 2026.

Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3DIS: Depth-driven decoupled image synthesis for universal multi-instance generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Hanshen Zhu, Zhen Zhu, Kaile Zhang, Yiming Gong, Yuliang Liu, and Xiang Bai. Training-free geometric image editing on diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19130–19140, 2025. doi: 10.1109/ICCV51701.2025.01778. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhu_Training-free_Geometric_Image_Editing_on_Diffusion_Models_ICCV_2025_paper.html.

# Appendix

## Contents


**A Extended Related Work . . . 17**

A.1 Geometric Conditions for Reference-Based Compositing . . . 17

A.2 Scene Synthesis and Geometry Control in Related Tasks . . . 18

A.3 Representation Prediction and Joint RGB-Depth Modeling . . . 19

**B Method Details . . . 20**

B.1 Notation and Denoising Objective . . . 21

B.2 Depth Injector: Architecture and Objective . . . 21

B.2.1 Encoder Corrections and Capacity . . . 21

B.2.2 Injector Width and Cost . . . 22

B.2.3 Feature Supervision and Structural Ablations . . . 22

B.2.4 Feature Targets Versus Decoded-Depth Targets . . . 23

B.2.5 Backbone Interfaces and Parameter Sharing . . . 24

B.3 Depth-Conditioned Rendering: Training and Inference . . . 24

B.3.1 Canvas and Fixed Observations . . . 24

B.3.2 Training Objectives and Depth Selection . . . 25

B.3.3 Training and Inference Algorithms . . . 25

B.4 Computational Cost . . . 25

**C Experimental Protocols and Further Results . . . 27**

C.1 Datasets and Evaluation Protocols . . . 27

C.1.1 Paired Data and Depth Targets . . . 27

C.1.2 Dataset Quality Control Prompt . . . 28

C.1.3 Unpaired Evaluation and Identity Margins . . . 30

C.1.4 Baseline Evaluation Protocol . . . 30

C.2 Depth Backbones and Training Protocol . . . 31

C.2.1 Depth Backbones and Training Settings . . . 31

C.2.2 Feature Correction Versus Diffusion-Based Composite Depth Generation . . . 31

C.3 Rendering Backbones and Ablation Protocols . . . 33

C.3.1 Backbones and Prompt Conditioning . . . 33

C.3.2 Training and Sampling Settings . . . 34

C.3.3 Comparison Across Diffusion Backbones . . . 34

C.3.4 Loss Ablations and Sampling Comparisons . . . 34

C.4 Evaluation Metrics . . . 35

C.4.1 Depth Alignment and Geometric Metrics . . . 35

C.4.2 Paired Uncertainty and Regional Evaluation . . . 36

C.4.3 Cross-Scorer Evaluation . . . 37

C.4.4 Photometric and Identity Metrics . . . 37

C.5 Limitations . . . 38

**D Additional Qualitative Results . . . 40**

# A Extended Related Work

## A.1 Geometric Conditions for Reference-Based Compositing

Table A1: Geometry Conditions in Reference-Based Compositing.

| Method | Composite-Scene Depth | Predicted, Not Assembled | Scene-Wide Prediction | 2D Inputs Alone | Fixed Geometry Before RGB |
|---|---|---|---|---|---|
| **(a) Implicit Composite Depth (RGB Only)** | | | | | |
| Paint-by-Example CVPR'23 (Yang et al., 2023) | ✗ | ✗ | ✗ | ✓ | ✗ |
| AnyDoor CVPR'24 (Chen et al., 2024b) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Insert-Anything AAAI'26 (Song et al., 2026) | ✗ | ✗ | ✗ | ✓ | ✗ |
| Personalize-Anything AAAI'26 (Feng et al., 2026) | ✗ | ✗ | ✗ | ✓ | ✗ |
| SHINE ICLR'26 (Lu et al., 2026) | ✗ | ✗ | ✗ | ✓ | ✗ |
| InsertFuse arXiv'26 (Li et al., 2026) | ✗ | ✗ | ✗ | ✓ | ✗ |
| HiFi-Inpaint CVPR'26 (Liu et al., 2026) | ✗ | ✗ | ✗ | ✓ | ✗ |
| **(b-1) Depth from the Existing Scene** | | | | | |
| MimicBrush NeurIPS'24 (Chen et al., 2024a) | ✗ | ✗ | ✗ | ✓ | ✓ |
| **(b-2) Geometry from Additional Inputs or Assembly Rules** | | | | | |
| BIFRÖST NeurIPS'24 (Instruction Branch) (Li et al., 2024a) | ✓ | ✗ | ✗ | ✗ | ✓ |
| DIRECT ICML'26 (Gong et al., 2026b) | ✗ | ✗ | ✗ | ✗ | ✓ |
| SpatialHand ICLR'26 (Wang et al., 2026) | ✓ | ✗ | ✗ | ✗ | ✓ |
| **(c) Predicted Composite-Scene Depth** | | | | | |
| Person-In-Situ CAVW'26 (Two-Stage Configuration) (Masuda et al., 2026) | ✓ | ✓ | ✓ | ✗ | ✓ |
| **D2R (Ours)** | ✓ | ✓ | ✓ | ✓ | ✓ |

**Comparison Criteria.** Table A1 extends Tab. 1 for reference-based compositing and specialized insertion tasks. The first three criteria ask whether depth describes the intended composite, is predicted rather than assembled, and is predicted jointly across the scene. A full-frame map assembled from separate depths does not satisfy the third criterion. The fourth tests use of standard compositing inputs alone, including dataset-provided reference masks (App. B.3.1). The final criterion asks whether a geometric condition is fixed before RGB synthesis; this includes depth maps and DIRECT's rendered RGB proxy. Person-In-Situ is classified by its two-stage configuration; its rendered body-depth condition is additional to the standard compositing inputs.

**RGB Compositing Without Explicit Composite Depth.** Paint-by-Example (Yang et al., 2023) and AnyDoor (Chen et al., 2024b) preserve reference identity through image features. Insert-Anything (Song et al., 2026) places the reference and target on one canvas and adapts an inpainting transformer with LoRA (Hu et al., 2022). It provides our matched baseline because D2R retains this rendering interface while adding depth prediction and supervision. Other mechanisms include token substitution in Personalize-Anything (Feng et al., 2026), iterative identity control in SHINE (Lu et al., 2026), and high-frequency feature preservation in HiFi-Inpaint (Liu et al., 2026). InsertFuse (Li et al., 2026) maps the insertion mask and its distance transform to visual-token features, specifying the editing region in 2D. Related appearance-conditioned methods include ObjectStitch (Song et al., 2023), IMPRINT (Song et al., 2024), ObjectDrop (Winter et al., 2024), ObjectMate (Winter et al., 2025), Add-it (Tewel et al., 2025), CareCom (Chen et al., 2026), and

$A^2$-Edit (Zheng et al., 2026). Their features can encode shape and geometry implicitly. D2R makes scene-wide relative depth an explicit condition before RGB synthesis.

**Separating Object Structure and Appearance.** PAIR Diffusion (Goel et al., 2024) controls object-level editing with segmentation-based structure and image-encoder appearance features. Refacade (Huang et al., 2026) learns to remove source texture while retaining object geometry, then uses shuffled reference patches to transfer texture. These methods separate structure from appearance through different conditioning mechanisms. D2R instead predicts the completed scene's depth, including the inserted object's structure and spatial relationship to the background.

**Depth from the Existing Scene.** MimicBrush (Chen et al., 2024a) transfers reference features through a second U-Net and optionally uses depth estimated from the input background before masking the compositing region. Table 1 describes this depth-enabled configuration. Its depth map provides the pre-edit geometry, including pixels later masked. Zero-shot depth-aware editing (Parihar et al., 2025) decomposes supplied depth into layers that a user transforms, requiring depth beyond the three compositing inputs. Both methods start from existing geometry; neither predicts completed-scene depth from the standard compositing inputs.

**Geometry from Additional Inputs or Assembly Rules.** In its instruction-driven branch, BIFRÖST (Li et al., 2024a) predicts an object box and depth value from language, then rescales reference depth into the background map. Table A1 classifies this branch, whereas our quantitative comparisons use mask-based inpainting (App. C.1.4). Table 6 separately tests depth fusion in D2R.

Explicit pose and placement controls provide another way to specify the intended arrangement. DIRECT (Gong et al., 2026b) uses a rendered RGB proxy to specify object pose; SpatialHand (Wang et al., 2026) combines background depth with a user-selected placement depth in the insertion mask and a numerical orientation condition for 6DoF object insertion. Its depth assembly preserves nearer background occluders. FreeInsert (Zhang et al., 2025) and the dual-branch GeoEdit (He et al., 2026) reconstruct and manipulate an object already present. D2R uses only standard compositing inputs.

**Geometric Editing and Cross-Image Compositing.** FreeFine (Zhu et al., 2025) separates object transformation, source-region inpainting, and target-region refinement without training. Its cross-image extension refines coarse RGB composites assembled from source objects and source/target masks, so its scope also includes compositing. The distinction from D2R is how the target structure is obtained: FreeFine transforms and refines RGB content, whereas D2R learns to predict scene-wide composite depth before RGB generation.

**Depth-First Human Compositing.** Person-In-Situ (Masuda et al., 2026) predicts a scene's depth with the reference person inserted, then generates the RGB composite. Its two-stage configuration uses rendered 3D body depth, estimated scene depth, and a body-derived box mask alongside scene and reference images. It fine-tunes diffusion models on depth and RGB targets constructed from video frames. D2R instead uses a general monocular depth prior, learning reference-conditioned feature corrections while keeping the estimator frozen. Encoder features from paired completed scenes supervise the injector. This design requires no human-body representation and supports the diverse object types in our evaluation. The distinction concerns the inputs and learned representation, rather than the depth-before-RGB ordering.

Our paired evaluation covers 107 distinct object tags (87 ID and 20 OOD; Tab. C2), with leading aggregate results on both splits. Among the 426 cases, 4 carry the person tag (0.9%), whereas 49 carry accessory or garment tags (11.5%). These counts use the literal `person_person`, `accessory_*`, and `garment_*` tags, without manually regrouping other objects. The unpaired benchmarks do not provide reference-category tags, so identifying a person-only subset would require manual annotation. We therefore provide a conceptual comparison with Person-In-Situ.

### A.2 Scene Synthesis and Geometry Control in Related Tasks

**Geometry-Guided Scene Synthesis.** Compose-and-Conquer (Lee et al., 2024) conditions RGB synthesis on foreground/background depth and exemplar semantics. 3DIS (Zhou et al., 2025) predicts scene depth from instance layouts before rendering RGB. These methods establish depth-guided synthesis as a prior design choice. Table A2 distinguishes their geometry inputs and adaptation from D2R, which predicts composite depth from an existing background, reference object, and 2D compositing mask.

Table A2: Geometry Inputs and Adaptation in Scene Synthesis.

| Method | Task and Inputs | Geometry Condition | Learned Components |
|---|---|---|---|
| Compose-and-Conquer ICLR'24 (Lee et al., 2024) | Composable synthesis from text, foreground/background depth, and exemplar semantics | Supplied foreground/background depth maps | Local/global fusers and cloned diffusion blocks; original diffusion backbone frozen |
| 3DIS ICLR'25 (Zhou et al., 2025) | Multi-instance generation from text and instance layouts | Scene depth generated before RGB | Fine-tuned depth generator and layout adapter; pretrained depth control and detail rendering without additional training |
| **D2R (Ours)** | Reference-based compositing from background, reference, and 2D compositing mask | Composite depth predicted before RGB | Depth-feature injector with frozen estimator; renderer LoRA |

**Geometry from Explicit Object Transformations.** Diffusion Handles (Pandey et al., 2024) lifts diffusion activations into 3D using estimated source depth, applies a user-specified transform, and reprojects the activations to guide RGB sampling. The effects-sensitive GeoEdit (Zhang et al., 2026) constructs transformed object images and masks, using mesh reconstruction for rotation, then performs in-context RGB inpainting with a trained FLUX.1-Fill model. It is distinct from the dual-branch GeoEdit (He et al., 2026) discussed above. These approaches use explicit transformations to specify the intended arrangement. D2R predicts that arrangement's relative depth from a separate reference, background, and compositing mask, without requiring a 3D transform.

**Depth-Conditioned Multi-View Canvases.** SIGNeRF (Dihlmann et al., 2024) edits NeRF scenes through depth-conditioned diffusion over a multi-view reference sheet. Geometry comes from the existing 3D scene and a selected region or external proxy mesh; the edited views are then used to update the NeRF. Its canvas organizes multiple views of supplied geometry. D2R's canvas instead combines a reference, masked background, and predicted composite depth for a single output image. The shared use of a canvas therefore does not imply the same source of geometric conditioning.

**Geometry Control During Sampling.** FreeInpaint (Gong et al., 2026a) optimizes initial noise and intermediate latents without updating model weights. Closed-loop test-time control (Zhao et al., 2026) adjusts conditioning using feedback from generated outputs. These methods modify RGB generation at inference, whereas D2R predicts composite depth once and holds it fixed throughout RGB sampling. Their published tasks differ from reference-preserving compositing, so we compare compositing-specific alternatives in Tab. 6: Latent Optimization toward Stage-1 depth and Joint Denoising of depth and RGB initialized with background depth. Appendix C.3 specifies both baselines. These results concern the evaluated implementations, rather than iterative control in general.

### A.3 Representation Prediction and Joint RGB-Depth Modeling

**Predictive Features and Joint RGB-Depth Models.** Table A3 separates representation choice from task. VGGT-World (Sun et al., 2026) predicts future geometry features from video observations and decodes them through frozen geometry components. D2R instead predicts features of a requested static composite from a background, reference object, and compositing mask. JointDiT (Kwon et al., 2025) and Modality Forcing (Duisterhof et al., 2026) support both joint RGB-depth modeling and depth-conditioned RGB generation. To test the joint-generation alternative in reference-based compositing, we implement a joint depth/RGB denoising baseline using D2R's data and rendering setup (Tab. 6; App. C.3.4). This is a task-specific comparison, rather than a reproduction or ranking of those models. D2R focuses on predicting the required depth condition from the compositing inputs before RGB generation.

**Feature Injection During RGB Sampling.** Plug-and-Play Diffusion Features (Tumanyan et al., 2023) preserves a guidance image's structure by injecting its spatial features and self-attention information during diffusion sampling, without training. D2R learns residual corrections inside a

Table A3: Prediction Targets and Adaptation Across Geometry Models.

| Method | Conditioning and Task | Prediction Target | Learned Components |
|---|---|---|---|
| VGGT-World ECCV'26 (Sun et al., 2026) | Video history for future geometry forecasting | Future geometry tokens, decoded into depth and point maps | Temporal flow transformer; VGGT encoder, remaining blocks, and 3D heads frozen |
| JointDiT ICCV'25 (Kwon et al., 2025) | Text, with optional RGB or depth for conditional generation | Joint or conditional RGB/depth latents | Depth-branch LoRA and cross-branch modules; pretrained FLUX backbone frozen |
| Modality Forcing arXiv'26 (Duisterhof et al., 2026) | Text, with optional RGB or depth for conditional generation | RGB latents and pixel-space depth | Post-trained DiT with modality-specific modules and noise levels; supports sparse depth supervision |
| **D2R (Ours)** | Background, reference, and 2D compositing mask for compositing | Completed-scene encoder features, decoded into composite depth | Reference-conditioned injector; depth encoder and decoder frozen, followed by renderer LoRA |

frozen depth estimator, using encoder features of paired completed scenes as targets. Feature injection is therefore a shared mechanism, while the learned prediction target and the network receiving the corrections differ.

**Predicting and Aligning Representations.** I-JEPA (Assran et al., 2023) predicts target-region representations from image context, while REPA (Yu et al., 2025) aligns diffusion features with representations from a pretrained visual encoder. These objectives provide precedents for feature-level supervision. D2R instead targets the depth-encoder features of a paired completed scene, predicting them from the background, reference, and mask before decoding them with the same estimator's unchanged decoder. Its supervision acts inside the depth predictor rather than aligning the RGB generator's hidden states.

**Encoder Adaptation and RGB Conditioning.** ViT-Adapter (Chen et al., 2023) and ViT-CoMer (Xia et al., 2024) augment visual encoders for dense prediction. D2R complements these architectural mechanisms by targeting features of the completed scene while leaving the dense decoder (Ranftl et al., 2021) unchanged. Rendering conditions can enter through dedicated branches (Zhang et al., 2023; Mou et al., 2024; Zhao et al., 2023; Ju et al., 2024), canvas patches (Tan et al., 2025), or image-embedding adapters (Ye et al., 2023). D2R supplies its predicted depth as a fixed patch on the reference-guided inpainting canvas (Song et al., 2026) and trains only renderer LoRA. Merged LoRA adds no renderer parameters.

**Consistency Feedback for Conditional Generation.** ControlNet++ (Li et al., 2024b) trains conditional generation by comparing an input control with the condition recovered from a one-step denoised image through a frozen discriminative model. D2R uses a related principle for RGB-depth consistency. Its supervision remains the paired target depth $D$, including when Stage 1 supplies the conditioning map $D_{\text{in}}$, and it combines this loss with depth-patch denoising. This separates the accuracy of the training target from errors in the supplied condition (App. B.3).

# B Method Details

We detail the objectives and algorithms underlying Sec. 3. Throughout, $I_{\text{bg}}$, $I_{\text{ref}}$, and $M$ denote the background, mask-selected reference, and 2D editable-region mask; $I$ and $\hat{I}$ denote the paired ground-truth and generated composites. The estimator-derived target is $D = F(I)$, and Stage 1 predicts $\hat{D} = \Phi_\phi(I_{\text{bg}}, I_{\text{ref}}, M)$. Neither depth is sensor-measured. Roman $E, G$ denote the depth encoder and decoder, whereas $\mathcal{E}, \mathcal{D}$ denote the RGB autoencoder. We use $\odot$ for elementwise multiplication and $\mathbf{I}$ for identity covariance.

### B.1 Notation and Denoising Objective

A frozen autoencoder $(\mathcal{E}, \mathcal{D})$ provides latent codes for inpainting (Rombach et al., 2022). During training, $x_0 = \mathcal{E}(C_{\text{sup}})$ encodes the supervision canvas containing the ground-truth composite, reference, and target depth. A separate input canvas $C_{\text{in}}$ supplies the masked background, reference, and conditioning depth. Algorithm 2 defines both explicitly. One noise tensor and timestep are sampled for the entire supervision canvas:

$$\begin{aligned} x_t &= a(t)\, x_0 + b(t)\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \\ \big(a(t),\, b(t)\big) &= (1-t,\; t) \quad \text{or} \quad \big(\sqrt{\bar{\alpha}_t},\; \sqrt{1-\bar{\alpha}_t}\big), \end{aligned} \tag{B1}$$

The first pair is the rectified-flow interpolant of FLUX.1-Fill and Qwen-Inpainting (Wu et al., 2025)[2], following flow-based generation (Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024). The second is the variance-preserving schedule of SD2-Inpainting (Rombach et al., 2022) and SDXL-Inpainting (Podell et al., 2024), where $\bar{\alpha}_t$ is the cumulative signal coefficient. Normalized time runs from clean data ($t = 0$) toward noise. For discrete schedules, $t = k/K$ with $k \in \{0, \ldots, K-1\}$ and $K = 1000$; coefficients are read at $k$. Training samples $\text{logit}(t) \sim \mathcal{N}(0, 1)$ for rectified flow; variance-preserving backbones sample $k$ uniformly.

Under either corruption schedule, we use $\epsilon_\theta$ for the denoiser. Given $x_t$, $t$, and conditioning $c$, it predicts target $u$. For a set $\Omega'$ of latent positions,

$$\mathcal{L}_{\text{denoise}}(\Omega') = \mathbb{E}_{x_0, \epsilon, t}\, \frac{1}{N_{\text{ch}}\, |\Omega'|} \sum_{i \in \Omega'} \big\| \epsilon_\theta(x_t, t, c)_i - u_i \big\|_2^2, \tag{B2}$$

where $i$ indexes spatial latent positions, $N_{\text{ch}}$ is the channel count, and the squared Euclidean norm sums over channels. The target is velocity, $u = \epsilon - x_0$, for rectified flow and noise, $u = \epsilon$, for variance-preserving diffusion. The one-step clean-latent estimate is

$$\hat{x}_0 = \begin{cases} x_t - t\, \epsilon_\theta(x_t, t, c), & u = \epsilon - x_0, \\ \big(x_t - b(t)\, \epsilon_\theta(x_t, t, c)\big)/a(t), & u = \epsilon, \end{cases} \tag{B3}$$

Equation (B8) specializes the first case to the deployed renderer. Both denoising losses use Eq. (B2); they differ only in spatial support. The consistency loss also depends on this clean-latent reconstruction and the normalized timestep weight $w(t)$.

We write $c = \text{Cond}(C, m, I_{\text{ref}})$ for the backbone-specific inpainting condition constructed from input canvas $C$, latent mask $m$, and reference. The binary mask marks generated positions with one. This condition includes the encoded canvas and mask, Redux reference embeddings for FLUX.1-Fill, and the ControlNet path for Qwen-Inpainting (Wu et al., 2025); tensor construction follows each backbone's released interface. Training corrupts the entire supervision canvas $C_{\text{sup}}$ while retaining the separate input condition. At inference, the reference, depth, and unmasked background remain observed inputs and are restored after every update (Eq. (B6)). Rectified-flow sampling uses explicit Euler updates; variance-preserving backbones use their released deterministic multistep solvers. Neither adds noise after initialization.

### B.2 Depth Injector: Architecture and Objective

#### B.2.1 Encoder Corrections and Capacity

The frozen model $F = G \circ E$ consists of encoder blocks $B_1, \ldots, B_N$ and dense decoder $G$ (Ranftl et al., 2021). The set $\mathcal{K}$ contains the encoder taps read by the decoder. The initial sequence $\tilde{h}_{\text{bg}}^0$ contains the masked-background tokens entering the first block. At tap $\ell$, $h_{\text{bg}}^\ell = B_\ell(\tilde{h}_{\text{bg}}^{\ell-1})$ denotes features before the current correction, including corrections propagated from earlier taps. Reference features $h_{\text{ref}}^\ell$ and training-only target features $h_{\text{gt}}^\ell$ come from the reference and complete composite, respectively. Mask-conditioned background queries attend to the background and reference sequences, yielding the correction

$$\delta^\ell = W_{\text{out}}^\ell\big(\alpha^\ell \odot \text{Attn}(Q, K, V)\big), \qquad Q = W_q\big(h_{\text{bg}}^\ell + e(M)\big), \qquad (K, V) = W_{kv}\big[h_{\text{bg}}^\ell; h_{\text{ref}}^\ell\big], \tag{B4}$$

[2]Official inpainting checkpoint: Qwen-Image-ControlNet-Inpainting.

Table B1: Depth Injector Accuracy and Cost Across Widths.

| Width $w$ | Params | Vs. Frozen | Latency ↓ | VRAM ↓ | AbsRel ↓ | | Avg ↓ |
|---|---|---|---|---|---|---|---|
| | | | | | In-Distribution | Out-of-Distribution | |
| Frozen | 0 | — | 25.8 ms | 3.25 GB | 0.1168 | 0.1506 | 0.1337 |
| 224 | 35.2M | 10.5% | 42.4 ms | 3.50 GB | 0.0507 | 0.0776 | 0.0642 |
| 320 | 70.1M | 20.9% | 45.5 ms | 3.72 GB | 0.0463 | 0.0767 | 0.0615 |
| 384 | 100.0M | 29.8% | 45.8 ms | 3.91 GB | 0.0473 | 0.0765 | 0.0619 |
| 448 | 135.2M | 40.3% | 47.6 ms | 4.26 GB | 0.0466 | 0.0770 | 0.0618 |
| 512 | 175.6M | 52.4% | 48.9 ms | 4.37 GB | 0.0471 | 0.0770 | 0.0621 |
| **640 (Adopted)** | **272.4M** | **81.2%** | **56.5 ms** | **5.21 GB** | **0.0462** | **0.0751** | **0.0606** |
| 768 | 390.2M | 116.4% | 59.5 ms | 5.87 GB | 0.0463 | 0.0778 | 0.0621 |
| 896 | 529.2M | 157.8% | 67.6 ms | 6.54 GB | 0.0445 | 0.0755 | 0.0600 |

Here $Q, K, V$ are attention queries, keys, and values, and $\mathrm{Attn}$ denotes multihead attention. The input to $W_{kv}$ concatenates background and reference token sequences; the projection produces separate keys and values. Layer indices on $Q, K, V, W_q, W_{kv}$ are suppressed. The embedding $e(M)$ places the mask on the token grid, $\alpha^\ell$ is a learned gate, and the $W$ matrices are linear projections. The zero-initialized output projection makes all corrections initially vanish, recovering the frozen estimator on the same masked-background input. Each tap has independent injector parameters; their union $\phi$ is the only trainable part of Stage 1. Subsequent frozen blocks propagate each correction to the decoder taps above it.

Width $w$ is the internal projection dimension in Eq. (B4). For Depth Anything V2 ViT-L (Yang et al., 2024b), projections map the 1024-channel encoder features to $w$ channels and back. Each tap has seven attention blocks with eight heads, with separate parameters at encoder blocks 5, 12, 18, and 24. Increasing $w$ changes injector capacity without changing the frozen encoder dimensions; parameters grow approximately quadratically with $w$.

### B.2.2 Injector Width and Cost

The adopted injector uses width $w = 640$. To assess the accuracy and cost of different widths, Tab. B1 varies only $w$ while fixing the taps, attention blocks, heads, gate, data, optimizer, and batch size. All trained variants use final checkpoints after 50K training iterations. Frozen denotes Depth Anything V2 ViT-L without an injector. Full-frame AbsRel is measured before RGB synthesis, and Avg is the mean of ID and OOD. Params counts trainable injector parameters; Vs. Frozen gives their size relative to the frozen backbone. Inference cost is measured on one RTX PRO 6000 (96 GB) in bf16 at batch size one with $518 \times 518$ inputs.

Width 640 reaches ID and OOD AbsRel of 0.0462 and 0.0751 with 272.4M trainable parameters. Increasing width to 896 lowers ID error to 0.0445 but raises OOD error to 0.0755, while parameters increase to 529.2M. Latency rises from 56.5 to 67.6 ms and inference memory from 5.21 to 6.54 GB. Thus the additional capacity brings a small ID gain without improving OOD accuracy, supporting width 640 as the adopted balance of accuracy and cost.

### B.2.3 Feature Supervision and Structural Ablations

The paired composite's encoder features provide a target that the frozen decoder already maps to composite depth. Feature supervision asks the injector to recover this representation from the background and reference, while retaining the pretrained feature-to-depth mapping. Decoded-depth supervision constrains only the resulting map: different feature configurations can produce similar depth, so a close output match need not recover the paired composite's encoder representation.

Equation (2) compares corrected background features with detached composite features at each tap. Features have shape $B \times C \times p_h \times p_w$, where $B$ is batch size, $C$ is channel count, and $p_h, p_w$ are the spatial grid dimensions. For each sample $b$, $\sigma_b^\ell = \mathrm{std}(h_{\mathrm{gt},b}^\ell)$ reduces the channel and both spatial dimensions to one scalar. The mean absolute feature error uses those same dimensions and is divided by $\mathrm{sg}(\sigma_b^\ell) + 10^{-6}$. We average the normalized errors first over samples and then over decoder taps. The frozen decoder produces $\hat{D}$ without participating in feature-loss backpropagation.

For one sample, let $H^* = \{h^\ell_{\text{gt}}\}_{\ell\in\mathcal{K}}$ and $\hat{H} = \{\tilde{h}^\ell_{\text{bg}}\}_{\ell\in\mathcal{K}}$ be the target and corrected feature collections supplied to $G$. Treating each collection and depth map as a vector, assume that $G$ is $L$-Lipschitz in the $\ell_1$ norm on a region containing both collections. Then

$$\left\|G(\hat{H}) - G(H^*)\right\|_1 \le L \left\|\hat{H} - H^*\right\|_1.$$

This conditional relation motivates matching features that the frozen decoder can already interpret. We do not estimate the Lipschitz constant, and Eq. (2) averages per-tap, sample-normalized errors rather than the unnormalized norm above. It therefore yields no numerical bound on test depth error or local contact accuracy and does not establish generalization. Performance across estimators and held-out categories is assessed empirically (Tab. C4).

D2R applies residual corrections across the full token grid. The masked-token ablation restricts these corrections to the editable region to test whether object-local updates suffice (Sec. 5). Relative-depth prediction depends on the surrounding scene, so one possible explanation for the full-grid advantage is that object and background features must adapt together. This concerns their representation of relative depth; it does not imply that insertion physically moves unchanged background surfaces.

To test how scene and reference information enter the correction, we compare attention and gating variants with the frozen estimator and feature objective held fixed. Joint attention lets each background token access both scene context and reference features, following multimodal transformer designs (Wu et al., 2025). Relative to reference-only cross-attention, it lowers full-frame AbsRel by approximately 2% on ID and 1% on OOD at each variant's final checkpoint. The gate and zero-initialized projection follow related conditioning mechanisms (Qiu et al., 2025; Zhang et al., 2023; Li et al., 2023; Alayrac et al., 2022). Adding the gate lowers error by a further 1% and 3%, respectively; the ungated variant also varies more across checkpoints.

### B.2.4 Feature Targets Versus Decoded-Depth Targets

To isolate the effect of the supervision target, we start from D2R's width-640 injector on frozen Depth Anything V2 ViT-L and replace only the encoder-feature objective with a decoded-depth objective. Both variants retain the same injector architecture, frozen encoder and decoder, training data, optimizer, and batch size, and use final checkpoints after 50K training iterations (Tab. 5). The alternative uses the normalized $\ell_1$ depth discrepancy from the Stage-2 consistency objective (App. B.3): the same normalization $A$ is applied separately to predicted and target composite-depth maps, without a background-based scale-and-shift fit. We score full-frame composite depth before RGB synthesis on both paired splits.

The two objectives differ in where gradients enter the frozen estimator. Feature supervision compares decoder inputs directly. Decoded-depth supervision must propagate gradients through the decoder to the injector, retaining decoder activations for backpropagation even though the decoder weights remain fixed. Both variants deploy the same injector and depth estimator, so this difference affects training computation rather than inference cost.

Encoder-feature supervision lowers ID and OOD AbsRel by 30.9% and 31.4%, respectively, relative to decoded-depth-only supervision; OOD error falls from 0.1094 to 0.0751. With architecture and training iterations held fixed, the comparison supports feature targets for repurposing this frozen estimator. The gain also avoids the decoder backward pass required by the alternative. Table B2 reports the decoded-depth-only variant's regional errors, using the regions defined in App. C.4.2.

Table B2: Regional AbsRel with Decoded-Depth Supervision.

| Split | Interior | Boundary Band | Background |
|---|---|---|---|
| ID | 0.1081 | 0.2072 | 0.0522 |
| OOD | 0.1711 | 0.2422 | 0.0909 |

A separate ablation retains the feature objective and adds decoded-depth supervision to test whether the two targets are complementary. Relative to feature-only training, the combined objective lowers ID full-frame AbsRel by about 5% but raises OOD error by about 3%. Retaining decoder activations for backpropagation also reduces the largest single-GPU batch by about 30%. The ID/OOD tradeoff and memory cost favor feature-only training in this setting.

**Effect on RGB Geometry with the Same Renderer.** We test whether the Stage-1 supervision difference remains useful after RGB synthesis. The two existing Stage-1 checkpoints in Tab. 5

supply depth to the same trained FLUX.1-Fill renderer, without further training. Only the depth source changes: both conditions use the same background, reference, and mask for all 139 OOD cases, with identical sampling settings and the standard depth normalization, resizing, quantization, and canvas layout.

Table B3: From Stage-1 Supervision to Final RGB Geometry with the Same Renderer.

| Stage-1 Supervision | OOD AbsRel ↓ | |
|---|---|---|
| | Predicted Composite Depth | Rendered RGB Geometry |
| Decoded Depth Only | 0.1094 | 0.089 |
| Encoder Feature Only (D2R) | **0.0751** | **0.067** |

Table B3 connects the predictor comparison to final compositing: the feature-supervised depth yields lower rendered-image AbsRel under the same RGB renderer. Thus, its benefit extends beyond the intermediate depth map to geometric agreement in the generated image. This comparison measures the utility of each predictor for the deployed renderer, which was trained with the default depth-supply strategy. It does not compare renderers retrained for each predictor or separate geometric accuracy from compatibility with the renderer's training distribution.

#### B.2.5 Backbone Interfaces and Parameter Sharing

Depth Anything V1 (Yang et al., 2024a), V2 (Yang et al., 2024b), and V3 (Lin et al., 2026) expose four decoder taps on one token grid. Metric-Anything (Yang et al., 2026) uses a patch pyramid, requiring corrections to be broadcast before decoding. InfiniDepth (Yu et al., 2026) requires taps across token resolutions, while Depth Pro (Bochkovskii et al., 2025) receives corrections in its image encoder alongside a separate patch encoder. The design and objective are shared, with tap selection, mask placement, and feature dimensions adapted per backbone.

The weight-sharing ablation replaces the default backbone's independent tap-specific injectors with shared parameters, keeping width 640 and the other settings fixed (Sec. 5). Sharing reduces parameter count but executes the same number of attention blocks. The adopted design uses independent parameters so each tap can learn a correction suited to its encoder layer.

### B.3 Depth-Conditioned Rendering: Training and Inference

#### B.3.1 Canvas and Fixed Observations

We use the dataset-provided reference-object masks to select the source object and the target masks to specify the compositing region. AnyInsertion++ supplies masks that follow the target object's silhouette, whereas the unpaired benchmarks are evaluated with coarse box-shaped target masks. The evaluations therefore cover both detailed object masks and coarse region masks. Because the datasets differ, these results do not isolate the effect of mask precision.

Let $S$ be the composite patch's side length. The input canvas places the masked background beside a column containing the reference above the conditioning depth:

$$C = \left[\, I_{\mathrm{bg}} \odot (1 - M) \;\middle|\; (I_{\mathrm{ref}}; D_{\mathrm{in}}) \,\right] \in \mathbb{R}^{S \times 1.5S \times 3}, \qquad \Omega = \Omega_{\mathrm{comp}} \cup \Omega_{\mathrm{ref}} \cup \Omega_{\mathrm{depth}}, \tag{B5}$$

All evaluated compositing datasets provide a reference object mask that identifies which object to use among other objects and background pixels. We use this supplied mask to select the reference; no automatic reference selection is performed. The mask identifies source appearance, whereas $M$ marks the editable region in the target scene. The reference is tightly cropped to its object mask, centered on a white square whose side is 1.05 times the crop's longer side, and resized to its patch. Training and inference use identical padding.

In Eq. (B5), the vertical bar places patches side by side, while the semicolon stacks the reference above the depth. For FLUX.1-Fill, $S = 1024$ gives a $1024 \times 1536$ canvas with $S/2 \times S/2$ auxiliary patches; resizing and padding to these sizes are implicit. The single-channel relative depth $D_{\mathrm{in}}$ is replicated across three channels. The sets $\Omega_{\mathrm{comp}}$, $\Omega_{\mathrm{ref}}$, and $\Omega_{\mathrm{depth}}$ index the corresponding latent positions. The same labels identify image patches after decoding the canvas.

At inference, the latent mask $m$ maps $M$ to the composite patch's latent grid, with one marking generated positions and zero on the reference, depth, and unmasked background. Writing $x$ for the

current sampled latent and $x_{\text{obs}} = \mathcal{E}(C)$ for the encoded input canvas, the sampler restores observed positions after every update:

$$x \odot (1 - m) = x_{\text{obs}} \odot (1 - m) \quad \text{after every step of Alg. 1,} \tag{B6}$$

### B.3.2 Training Objectives and Depth Selection

Stage 2 uses three equally weighted losses. Over valid latent positions $\mathcal{V}$ excluding square padding, the two denoising losses read separate parts of one prediction:

$$\begin{aligned} &\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pixel}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{cons}}, \\ &\mathcal{L}_{\text{pixel}} = \mathcal{L}_{\text{denoise}}\big((\Omega_{\text{comp}} \cup \Omega_{\text{ref}}) \cap \mathcal{V}\big), \qquad \mathcal{L}_{\text{depth}} = \mathcal{L}_{\text{denoise}}\big(\Omega_{\text{depth}} \cap \mathcal{V}\big), \end{aligned} \tag{B7}$$

The two denoising losses supervise different patches of the same canvas prediction. Its depth target is always $D$, regardless of which depth map conditions the input. The auxiliary depth loss trains the shared denoiser to represent composite geometry. At inference, each sampling step restores the supplied depth condition, discarding the predicted depth-patch values.

To enforce RGB-to-depth consistency, the rectified-flow prediction also yields a one-step clean estimate and its decoded composite patch:

$$\hat{x}_0 = x_t - t\,\epsilon_\theta(x_t, t, c), \qquad \hat{I}_{\text{comp}} = \mathcal{D}(\hat{x}_0)\big|_{\Omega_{\text{comp}}}, \tag{B8}$$

The full consistency loss is $\mathcal{L}_{\text{cons}} = w(t)\|A(F(\hat{I}_{\text{comp}})) - \text{sg}(A(D))\|_1$. The target remains $D$ even when the input depth is imperfect, and $w(t) = 1 - t$ reduces the contribution of noisy early one-step estimates. Gradients flow through the frozen RGB decoder $\mathcal{D}$ and depth estimator $F$ to LoRA only.

Normalization $A$ is applied independently to each predicted and target depth map. For each sample, map $Z$ is flattened over the entire composite patch, including padding; the reference and depth-conditioning patches are excluded. We compute $\mu(Z) = \text{median}(Z)$ and $r(Z) = \text{mean}\,|Z - \mu(Z)|$, then normalize as $A(Z) = (Z - \mu(Z))/\max(r(Z), 10^{-6})$. The denominator floor handles nearly constant depth. Only after normalization do we apply the validity mask when averaging the loss; it does not affect the median or mean absolute deviation. Evaluation instead fits the background alignment in Eq. (C2).

To expose the renderer to prediction errors while keeping supervision fixed, we select the conditioning depth independently for each sample and iteration:

$$D_{\text{in}} = \begin{cases} \hat{D} & \text{with probability } 1/2, \\ D & \text{with probability } 1/2, \end{cases} \tag{B9}$$

The adopted selection policy changes the conditioning input while retaining $D$ in the supervision canvas. To test other ways of exposing the renderer to depth, the blending ablation instead supplies $D_{\text{in}} = \lambda\hat{D} + (1 - \lambda)D$, with $\lambda \sim \text{Beta}(0.2, 0.2)$ following mixup (Zhang et al., 2018a); the single-source ablations use $D_{\text{in}} = D$ (GT-only) or $D_{\text{in}} = \hat{D}$ (predicted-only). Selection supplies one intact map, whereas blending interpolates between the maps. When conditioning differs from $D$, simply reproducing the input depth need not minimize the supervised depth discrepancy. Stage-1 predictions are exported before Stage-2 training, so only the frozen consistency estimator runs inside its training loop. Appendix C.3 reports the comparison.

### B.3.3 Training and Inference Algorithms

Algorithm 1 specifies rectified-flow inference, and Alg. 2 gives one training step using the same $\text{Cond}$ notation. Stage 1 predicts $\hat{D}$ once; Stage 2 samples the masked composite region for $T$ steps and resizes the decoded composite to the background resolution. FLUX.1-Fill receives the reference through Redux; other backbones use the prompts in App. C.3. LoRA is merged before deployment.

## B.4 Computational Cost

Stage 1 adds a 272.4M-parameter injector to frozen Depth Anything V2 ViT-L (Yang et al., 2024b) (335.3M parameters), an 81.2% increase. Stage 2 trains rank-256 LoRA with scaling 256 while

**Algorithm 1** D2R Inference with Fixed Composite Depth.

**Require:** background $I_{\mathrm{bg}}$, mask-selected reference $I_{\mathrm{ref}}$, 2D editable-region mask $M$, frozen depth model $F = G \circ E$ with trained injector $\phi$, generator $\epsilon_\theta$ with merged LoRA, frozen autoencoder $(\mathcal{E}, \mathcal{D})$, timesteps $1 = t_1 > t_2 > \cdots > t_{T+1} = 0$

**Ensure:** composite $\hat{I}$ at the resolution of $I_{\mathrm{bg}}$

$\hat{D} \leftarrow G\big(\{\tilde{h}^{\ell}_{\mathrm{bg}}\}_{\ell \in \mathcal{K}}\big)$ with $\tilde{h}^{\ell}_{\mathrm{bg}}$ from Eq. (2) ▷ Stage 1, one forward pass

$C \leftarrow \big[\, I_{\mathrm{bg}} \odot (1 - M) \;\big|\; (I_{\mathrm{ref}}; \hat{D}) \,\big]$ ▷ canvas, Eq. (B5)

Construct latent mask $m$ from $M$ on $\Omega_{\mathrm{comp}}$, with zeros on the other patches

$x_{\mathrm{obs}} \leftarrow \mathcal{E}(C), \quad c \leftarrow \mathrm{Cond}(C, m, I_{\mathrm{ref}})$

$x \leftarrow \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$

**for** $k = 1, \ldots, T$ **do**

  $x \leftarrow x + (t_{k+1} - t_k)\, \epsilon_\theta(x, t_k, c)$ ▷ one rectified-flow step on the whole canvas

  $x \leftarrow m \odot x + (1 - m) \odot x_{\mathrm{obs}}$ ▷ clamp, Eq. (B6)

**end for**

$\hat{C} \leftarrow \mathcal{D}(x)$

**return** $\hat{I} \leftarrow \hat{C}\big|_{\Omega_{\mathrm{comp}}}$ resized to the resolution of $I_{\mathrm{bg}}$

**Algorithm 2** Stage-2 Training with RGB and Depth Supervision.

**Require:** paired sample $(I_{\mathrm{bg}}, I_{\mathrm{ref}}, M, I)$, target depth $D = F(I)$, exported Stage-1 depth $\hat{D}$, frozen $\mathcal{E}$, $\mathcal{D}$, $F$, denoiser $\epsilon_\theta$ with trainable LoRA $\theta$, configured optimizer

$D_{\mathrm{in}} \leftarrow \hat{D}$ with probability $0.5$ and $D$ otherwise ▷ per-sample selection, Eq. (B9)

$C_{\mathrm{in}} \leftarrow \big[\, I_{\mathrm{bg}} \odot (1 - M) \;\big|\; (I_{\mathrm{ref}}; D_{\mathrm{in}}) \,\big]$ ▷ input canvas

$C_{\mathrm{sup}} \leftarrow \big[\, I \;\big|\; (I_{\mathrm{ref}}; D) \,\big]$ ▷ supervision canvas, depth patch always $D$

Construct latent mask $m$ from $M$ on $\Omega_{\mathrm{comp}}$, with zeros elsewhere

$c \leftarrow \mathrm{Cond}(C_{\mathrm{in}}, m, I_{\mathrm{ref}}), \quad x_0 \leftarrow \mathcal{E}(C_{\mathrm{sup}})$

$\epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad \mathrm{logit}(t) \sim \mathcal{N}(0, 1), \quad x_t \leftarrow (1 - t)\, x_0 + t\, \epsilon$ ▷ rectified flow, Eq. (B1)

$v \leftarrow \epsilon_\theta(x_t, t, c)$ ▷ one forward pass over the whole canvas

$\mathcal{L}_{\mathrm{pixel}} \leftarrow \|v - (\epsilon - x_0)\|_2^2$ averaged on $(\Omega_{\mathrm{comp}} \cup \Omega_{\mathrm{ref}}) \cap \mathcal{V}$ ▷ Eq. (B7)

$\mathcal{L}_{\mathrm{depth}} \leftarrow \|v - (\epsilon - x_0)\|_2^2$ averaged on $\Omega_{\mathrm{depth}} \cap \mathcal{V}$

$\hat{x}_0 \leftarrow x_t - t\, v, \quad \hat{I}_{\mathrm{comp}} \leftarrow \mathcal{D}(\hat{x}_0)\big|_{\Omega_{\mathrm{comp}}}$ ▷ Eq. (B8)

$\mathcal{L}_{\mathrm{cons}} \leftarrow (1 - t)\, \big\| A(F(\hat{I}_{\mathrm{comp}})) - \mathrm{sg}(A(D)) \big\|_1$ ▷ Eq. (3), full form of App. B.3

Update $\theta$ with the configured optimizer using $\nabla_\theta \mathcal{L}_{\mathrm{total}}$ ▷ LoRA only; $\mathcal{E}, \mathcal{D}, F$ frozen

freezing the backbone, autoencoder, and Redux encoder. Adapter sizes are 927M for FLUX.1-Fill, 53M for SD2-Inpainting (Rombach et al., 2022), 372M for SDXL-Inpainting (Podell et al., 2024), and 1,887M for Qwen-Inpainting (Wu et al., 2025). Only the injector adds deployed parameters; renderer LoRA is merged into the backbone.

We measure inference on one NVIDIA RTX PRO 6000 (96 GB), at batch size one on one OOD sample. Each row of Tab. B4 runs in a separate process with one warm-up and five timed passes. Latency is the median of those passes, excluding disk I/O; memory is peak process allocation including weights and activations. Stage-1 timing covers both the frozen estimator and its injector.

Following each released Diffusers setup, SD2 and SDXL use fp16; the depth models, Qwen, and FLUX use bf16. Stage 1 receives $518 \times 518$ inputs. All Stage-2 renderers use 28 steps, with canvases of $512 \times 768$ for SD2, $1024 \times 1536$ for SDXL and FLUX, and $768 \times 1152$ for Qwen. Guidance is 7.0 for SD2 and SDXL, 4.0 for Qwen, and distilled guidance 30 for FLUX.

Two-stage latency adds the 56.5 ms Stage-1 pass to the measured renderer latency and is rounded to two decimal places. Inference memory peaks are reported per process; summing them would not measure the peak of a deployment with both stages resident. The table therefore separates the cost of predicting geometry from that of RGB generation.

The matched Insert-Anything* (Song et al., 2026) baseline shares D2R's backbone, canvas, Redux embedding, and LoRA configuration. D2R therefore adds one Stage-1 pass to otherwise matched RGB generation cost. This does not compare against released Insert-Anything's full-height reference canvas, whose token count differs.

Table B4: Inference Cost on RTX PRO 6000.

| Component | Latency ↓ | Two-Stage Latency ↓ | Inference VRAM ↓ | Canvas (Height × Width) | Trainable Parameters |
|---|---|---|---|---|---|
| Frozen DAv2 NeurIPS'24 | 25.8 ms | — | 3.25 GB | 518 × 518 | 0 |
| + Depth Injector (Stage 1; Sec. 3.1) | 56.5 ms | — | 5.21 GB | 518 × 518 | 272.4M |
| SD2-Inpainting (Stage 2) | 0.81 s | 0.87 s | 4.39 GB | 512 × 768 | 53M (LoRA) |
| SDXL-Inpainting (Stage 2) | 2.32 s | 2.38 s | 13.16 GB | 1024 × 1536 | 372M (LoRA) |
| Qwen-Inpainting (Stage 2) | 14.24 s | 14.30 s | 65.93 GB | 768 × 1152 | 1,887M (LoRA) |
| FLUX.1-Fill (Stage 2; Sec. 3.2) | 22.34 s | 22.40 s | 38.50 GB | 1024 × 1536 | 927M (LoRA) |

Table C1: AnyInsertion++ Dataset Design.

| Property | Original AnyInsertion | **AnyInsertion++ (Ours)** |
|---|---|---|
| Paired Test Cases | 38 | 287 ID and 139 OOD |
| Held-Out Object Tags | Not separated | 21 tags, OOD split only |
| Training Pool Examined | 77,745 before curation | 52,918 kept, 24,827 removed |
| Ground-Truth Composite Depth | None | One per paired sample |
| Matched Comparator | None | Insert-Anything retrained on ours |

On NVIDIA RTX PRO 6000, the width-640 depth injector reaches a peak training memory of 72.4 GiB. Stage-2 peaks per device are 81.3 GiB for SD2, 67.8 GiB for SDXL, 94.0 GiB for Qwen, and 66.8 GiB for FLUX. These device-level readings are converted from MiB to GiB and correspond to the configurations in Apps. C.2 and C.3. Batch size and adapter settings differ across backbones, so these values document the memory needed by each training setup rather than isolate the backbone's memory efficiency.

# C Experimental Protocols and Further Results

## C.1 Datasets and Evaluation Protocols

### C.1.1 Paired Data and Depth Targets

AnyInsertion++ curates AnyInsertion (Song et al., 2026) for training and paired evaluation, excluding samples with annotation errors in automatically produced backgrounds or masks. The checks below retain 52,918 of the 77,745 training samples. Evaluation expands from the original 38 test cases to 287 ID and 139 OOD cases, each with an estimator-derived depth target (Tab. C1). The category split, annotation checks, and depth preparation are described below.

We use Claude Opus 5[3] to check annotation validity for all 77,745 training samples with one fixed English prompt. It receives the background, reference image, dataset-provided reference object mask, target mask, and ground-truth composite, in that order. The checks cover target-object presence, target-mask alignment, reference-mask validity, object-instance identity, and agreement between the background and the composite with that object absent. Aesthetics, photographic quality, and difficulty are excluded, so difficult but correctly labelled samples are retained.

The model returns a confidence score from 0 to 10, a usability label, and the first failed check. Human inspection of 300 randomly sampled images per score tier informed the threshold; a sample was judged usable when at least two of three annotators agreed. Excluding scores below 8 removes 24,827 samples and retains 52,918. Appendix C.1.2 reproduces the exact prompt.

The category split tests generalization beyond the compositing training set. The corpus contains 109 object tags grouped into categories: the 287-case ID split shares training categories, whereas

[3] Anthropic, Introducing Claude Opus 5.

the 139-case OOD split contains categories absent from training. Both include insertion and replacement. The split construction holds out 21 tags; Tab. C2 lists those with scored samples. A compositing case is the evaluation unit, but related insertion and replacement cases can share an image identifier. This gives 155 ID and 82 OOD image groups, for which App. C.4.2 reports image-cluster uncertainty alongside case-level uncertainty.

**Train/Test Image Overlap.** We compare all 287 ID and 139 OOD test images against the 52,918 training samples. An exact thumbnail match means identical SHA-1 hashes of $64 \times 64$ grayscale thumbnails. For near duplicates, a 64-bit DCT-based perceptual hash selects candidates with Hamming distance at most 8; normalized cross-correlation of at least 0.95 between the $64 \times 64$ thumbnails confirms a match. Neither test split contains a thumbnail match or a confirmed near-duplicate image under these criteria (0/287 ID; 0/139 OOD).

Ground-truth composite depth is the estimator-derived target $D = F(I)$, not sensor ground truth. The default frozen Depth Anything V2 ViT-L (Yang et al., 2024b) runs offline at 518-pixel input resolution. Its relative inverse depth, with larger values denoting nearer surfaces, is independently min-max normalized to $[0, 1]$, resized to the original image, and stored as an 8-bit single-channel PNG. These maps supply the Stage-2 target patch and default geometric reference; Stage 1 instead learns from the same estimator's encoder features. Background and reference depth maps for the alternative provisioning strategies use the same extraction procedure.

At inference, Stage-1 predictions $\hat{D}$ use only the three compositing inputs. They undergo the same normalization and are rounded from a 16-bit intermediate to 8-bit PNGs for the depth patch, including on unpaired evaluation data. Thus *GT Composite Depth* denotes the paired-image target, while deployed conditioning uses predicted composite depth. Scorer dependence is tested in App. C.4.

To test whether improvements extend beyond rigid objects, we group garment, accessory, person, animal, food, and plush tags as non-rigid and the remaining tags as rigid. This operational rule groups categories rather than assigning physical rigidity to every instance. Table C2 lists 146 rigid and 141 non-rigid ID cases from 42 and 45 tags, and 119 rigid and 20 non-rigid OOD cases from 14 and 6 tags. The OOD result is therefore dominated by rigid objects, with a smaller non-rigid sample.

D2R lowers AbsRel in all four groups (Tab. C3). Replacing predicted depth with depth from the paired composite further tests the benefit of a better condition. That comparison keeps the renderer fixed; it does not uniquely attribute the remaining error to Stage 1.

### C.1.2 Dataset Quality Control Prompt

The same prompt and image order are used for every training sample. Samples scoring at least 8 are retained; the three usability labels and score intervals are defined in the prompt.

```
You are auditing training samples for an object-insertion dataset. Each sample is
meant to be a matched set: a background image, a reference image containing one
object, a mask marking that object in the reference, a mask marking where the
object belongs in the target, and a ground-truth composite in which that object
has been inserted at that location. The masks and backgrounds in this release
were produced automatically, so some samples are mislabelled.

You receive five images per sample, in this order:
  1. image_src  - the background before insertion
  2. image_ref  - the image the object is taken from
  3. mask_ref   - the mask of the object inside image_ref
  4. mask_tar   - the mask of the region the object should occupy
  5. image_tar  - the ground-truth composite

Judge only whether the sample is correctly labelled. Do not judge aesthetics,
photographic quality, or difficulty. A sample that is hard but correctly
labelled is a good sample.

Check, in order:
  1. Does image_tar contain an object inside the mask_tar region? A mask over
     empty background, or over the wrong part of the scene, is an error.
  2. Does mask_tar follow that object's silhouette? A clearly different shape,
```

Table C2: **Object Tags of the Two Test Splits by Rigidity Group.** Bands (a) and (b) list ID and OOD tags with scored-sample counts. Groups follow the category rule in the text.

| Group | Tags (Count) |
|---|---|
| **(a) In-Distribution (ID) Test Split: Categories Overlap the Training Set** | |
| Rigid ($n = 146$, 42 tags) | ashcan (4), balloons (2), bed (2), book (4), bottle (4), box (4), cabinet_air_conditioner (4), ceiling_lamp (2), chair (2), coat_rack (2), flashlight (4), flower (4), hair_dryer (4), hammer (4), jug (4), ladder (4), lamp (4), laptop (4), lipstick (2), microwave (4), motorcycle (2), mug (2), paper_box (4), percussion (4), pillow (4), pistol (4), projector (4), rifle (2), rockery (4), sink (4), soccer (4), sofa (4), spoon (4), strings (4), sunflower (2), table (4), toilet (4), toy_figure (4), tripod (4), walking_stick (4), wall_lamp (2), wind_instruments (4) |
| Non-Rigid ($n = 141$, 45 tags) | accessory_bag (2), accessory_belt (2), accessory_bracelet (2), accessory_cup (3), accessory_dress (2), accessory_glasses (2), accessory_glove (2), accessory_hat (2), accessory_necklace (3), accessory_pendant (3), accessory_scarf (2), accessory_shoe (2), accessory_trousers (2), accessory_wallet (2), accessory_watch (2), apple (4), bagged_snacks (2), banana (4), broccoli (4), cabbage (2), chili (4), chinese_cabbage (4), chinese_hot_dishes (4), coconut (4), corn (4), cucumber (4), eggplant (4), momordica_charantia (4), onion (4), pear (4), pineapple (4), pizza (4), plush_toy (2), potato (4), pumpkin (4), roast_duck (4), scarf (2), sweet_potato (4), tomato (4), toy_bear (4), toy_cat (4), toy_dog (2), watermelon (4), winter_melon (2), person_person (4) |
| **(b) Out-of-Distribution (OOD) Test Split: No Category Overlap with the Training Set** | |
| Rigid ($n = 119$, 14 tags) | desk_air_conditioner (14), fork (10), hanger (18), lamp_post (2), mirror (6), piano (6), picture_frame (20), refrigerator (18), remote_control (6), ring (2), scissors (10), spanner (2), toy (3), wheelchair (2) |
| Non-Rigid ($n = 20$, 6 tags) | garment_jacket (5), garment_shirt (3), garment_short (3), garment_skirt (5), cat (2), dog (2) |

Table C3: **Per-Group Results by Object Rigidity.** Full-frame rendered-image AbsRel and PSNR compare D2R with the listed baselines within the groups in Tab. C2. D2R uses predicted composite depth; the gray row replaces it with depth from the paired composite and is excluded from rankings. Bold marks the best ranked AbsRel.

| Method | In-Distribution | | | | Out-of-Distribution | | | |
|---|---|---|---|---|---|---|---|---|
| | Rigid | | Non-Rigid | | Rigid | | Non-Rigid | |
| | AbsRel ↓ | PSNR ↑ | AbsRel ↓ | PSNR ↑ | AbsRel ↓ | PSNR ↑ | AbsRel ↓ | PSNR ↑ |
| D2R (Ground-Truth Depth) | 0.041 | 21.7 | 0.047 | 23.7 | 0.059 | 25.0 | 0.060 | 23.5 |
| **D2R (Predicted Depth)** | **0.045** | 21.2 | **0.050** | 23.2 | **0.067** | 24.3 | **0.071** | 22.3 |
| Insert-Anything* AAAI'26 (Song et al., 2026) | 0.059 | 20.1 | 0.066 | 22.3 | 0.116 | 21.8 | 0.133 | 20.4 |
| DIRECT ICML'26 (Gong et al., 2026b) | 0.082 | 18.6 | 0.094 | 20.2 | 0.134 | 19.6 | 0.154 | 18.5 |
| Nano Banana 2 Tested 2026-08 | 0.192 | 15.0 | 0.158 | 17.3 | 0.254 | 15.8 | 0.136 | 20.3 |

```
    or a mask covering a different object, is an error. A loose or non-convex
    mask that still follows the object is not an error.
  3. Does mask_ref mark a real object in image_ref rather than background or
    empty space?
  4. Are the object in image_ref and the object inside mask_tar in image_tar the
    same instance? A different object of the same category is an error.
  5. Is image_src the same scene as image_tar with that object absent? A
    different scene, or one where the object is still present, is an error.

Return one JSON object and nothing else:
  {"score": <integer 0-10>, "usable": "yes" | "partial" | "no",
   "reason": "<one short clause>"}

score is your confidence that the sample passes all five checks, where 10 means
every check passes cleanly and 0 means the sample is unusable.
usable is "no" for score 0-4, "partial" for 5-7, and "yes" for 8-10.
reason names the first check that failed, or "clean" when none did.
```

### C.1.3 Unpaired Evaluation and Identity Margins

TF-ICON (Lu et al., 2023), AIComposer (Li et al., 2025), and ComplexCompo (Lu et al., 2026) provide 332, 272, and 300 evaluation cases without paired composites. We therefore compare generated objects with their references to measure reference retention. Predictions use target-box crops, references use dataset-provided object masks, and both are placed on white. All methods use the same complete case set for each benchmark, with scores averaged over its cases.

CLIP-I uses IsoCLIP (Magistri et al., 2026) with a SigLIP2 (Tschannen et al., 2025) backbone; DINO-I uses the [CLS] token of DINOv3 ViT-L/16 (Siméoni et al., 2025). Paired and unpaired evaluation share the encoder settings in App. C.4.4, but use different comparison targets and crops. Object scale and residual background can affect embedding similarity, so these scores assess reference agreement rather than scene geometry. D2R uses FLUX.1-Fill with injector-predicted Depth Anything V2 ViT-L depth; Insert-Anything* (Song et al., 2026) leaves the depth patch empty.

To interpret the identity margins, let $S_b$ be the highest similarity among the fifteen baselines for a given benchmark and encoder, and $S_{\mathrm{D2R}}$ the D2R score in Tab. 4. Cosine distance to the reference is $d = 1 - S$; its relative reduction is $100(S_{\mathrm{D2R}} - S_b)/(1 - S_b)$ percent. Matched Insert-Anything* is the strongest baseline in all six comparisons.

We aggregate benchmarks by averaging their cosine distances with equal weight, then compute $100(\bar{d}_b - \bar{d}_{\mathrm{D2R}})/\bar{d}_b$. Mean CLIP distance falls from 0.215 to 0.096 (55.4%), and mean DINO distance from 0.268 to 0.208 (22.6%); the CLIP reduction rounds to 55% in the abstract. For example, on TF-ICON, CLIP-I rises from 0.760 to 0.902, corresponding to a distance reduction from 0.240 to 0.098 (59.1%). Scores and distances are displayed to three decimal places, while percentages use the four-decimal evaluation records.

DINO cosine distance decreases by 32.6% on TF-ICON, 9.4% on AIComposer, and 20.7% on ComplexCompo. AIComposer gives the smallest margin: D2R scores 0.813 versus 0.794 for Insert-Anything*. These percentages measure changes in embedding agreement; they do not estimate identity accuracy or the frequency of compositing failures.

### C.1.4 Baseline Evaluation Protocol

Eleven open-source methods use released checkpoints and, unless specified below, official default settings: Paint-by-Example (Yang et al., 2023), TF-ICON (Lu et al., 2023), AnyDoor (Chen et al., 2024b), MimicBrush (Chen et al., 2024a), BIFRÖST (Li et al., 2024a), DreamFuse (Huang et al., 2025), UniCombine (Wang et al., 2025), Personalize-Anything (Feng et al., 2026), SHINE (Lu et al., 2026), DIRECT (Gong et al., 2026b), and HiFi-Inpaint (Liu et al., 2026).

**BIFRÖST.** We run mask-based inpainting from the official repository's main branch with default settings, using `bifrost.ckpt` from `lingxiao2049/Bifrost`.

**DIRECT.** We adapt DIRECT (Gong et al., 2026b) to automatic evaluation by selecting the 3D proxy rotation with the highest silhouette IoU against the target mask. The dataset-provided reference mask becomes the reference image's alpha channel, and `TRELLIS-image-large` reconstructs a 3D Gaussian proxy from this RGBA input without a background-removal model. Only rotation is optimized: the proxy retains its native TRELLIS scale, with no scaling operation or scale search, and translation is reset to zero for every candidate rather than accumulated across trials.

The initial grid evaluates $12 \times 5 \times 2 = 120$ unique rotations: $y \in \{0°, 30°, \ldots, 330°\}$, $x \in \{-60°, -30°, 0°, 30°, 60°\}$, and $z \in \{0°, 180°\}$. Each of the three highest-IoU candidates is refined with eight random rotation proposals at each angular scale of $15°$, $6°$, and $2°$. A refinement is retained only when its best proposal improves IoU. The highest-IoU final candidate is chosen after $120 + 3 \times 24 = 192$ orientation evaluations.

For the matched comparison, we retrain Insert-Anything (Song et al., 2026) on AnyInsertion++ as Insert-Anything*; its released checkpoint is excluded because the original training corpus can overlap our evaluation splits. The retrained baseline shares D2R's data, backbone, canvas, Redux embedding, and LoRA configuration, but uses an empty depth patch and only the RGB denoising loss. Both renderers use their final checkpoints after 100K training iterations and sample with 28 steps, guidance 30, and seed 0.

**Commercial Editors.** We queried GPT-Image-1, GPT-Image-2[4], and Nano Banana 2[5] through public interfaces in August 2026. Each receives the same background image, reference image, and target mask with the ADD or REPLACE instruction below. Each request produces one image, and every evaluation case has one valid output. We do not choose among valid outputs, so every editor is scored on the same cases. Unspecified options use official defaults.

GPT-Image-1 and GPT-Image-2 use `quality="high"` with available output sizes of $1024\times1024$, $1536 \times 1024$, and $1024 \times 1536$. Nano Banana 2 uses `gemini-3.1-flash-image-preview` with `image_size="1K"`. We select the available aspect ratio closest to the input scene. If the returned dimensions differ, bicubic interpolation resizes the entire output directly to the original input-scene dimensions, without cropping or padding; outputs already at the input resolution remain unchanged. This retains the full generated frame and avoids content-dependent choices of crop or padding placement. Identity scoring follows App. C.1.3.

The following instructions are reproduced verbatim:

> **ADD.** I have a reference image of a product/object, a target image, and a mask indicating the target location. Please add the product from the reference image into the masked area of the target image in a natural and realistic way. Keep the original scene, structure, and background intact. The added product should look like it naturally belongs in the scene.
>
> **REPLACE.** I have a reference image of a product/object, a target image, and a mask indicating the area to edit. Please replace the existing object within the masked area of the target image with the product from the reference image. Keep the surrounding scene, pose/structure, background, and lighting exactly the same. Only change the object in the masked area to match the reference.

## C.2 Depth Backbones and Training Protocol

### C.2.1 Depth Backbones and Training Settings

Table C4 evaluates Depth Anything V1 (Yang et al., 2024a), V2 (Yang et al., 2024b), and V3 (Lin et al., 2026) with ViT-S, ViT-B, and ViT-L; Depth Pro (Bochkovskii et al., 2025) with ViT-L; Metric-Anything (Yang et al., 2026) with ViT-H/16+; and InfiniDepth (Yu et al., 2026) with ViT-L/16. Each uses its released resolution, including 518 pixels for Depth Anything. Appendix B.2 describes attachment interfaces. Each backbone supplies its own supervision and scoring targets; results therefore measure injection gains within that backbone.

Stage 1 trains with Eq. (2) on the curated 52.9K pool in bf16 on NVIDIA RTX PRO 6000 (96 GB) GPUs. All Stage 1 ablations use final checkpoints after 50K training iterations, including the backbone configurations in Tab. C4 and the supervision comparisons in App. B.2.4. Most configurations use batch size 24, or 1.2M sample presentations. Depth Pro (Bochkovskii et al., 2025) uses batch size 2 and Metric-Anything (Yang et al., 2026) batch size 1, corresponding to 100K and 50K sample presentations. Thus equal iteration counts do not imply equal sample exposure.

The default width-640 injector on Depth Anything V2 ViT-L (Yang et al., 2024b) uses $518 \times 518$ inputs, batch size 24, and Prodigy (Mishchenko & Defazio, 2024). Its median step time is 1.25 s, and training takes approximately 17.5 h. Appendix B.4 reports training memory.

Adding the injector lowers AbsRel and raises $\delta_1$ in every within-estimator comparison in Tab. C4. The approach therefore transfers across the tested pretrained depth representations. Because targets and sample exposure differ across estimators, the absolute scores do not provide a controlled architecture ranking. Appendix B.2 reports the structural ablations.

### C.2.2 Feature Correction Versus Diffusion-Based Composite Depth Generation

This experiment asks whether to predict composite depth by correcting a frozen estimator's features or by directly generating a depth image with diffusion. The latter follows the depth-generation strategy used in 3DIS (Zhou et al., 2025), which conditions a diffusion model on text and instance

[4]OpenAI model documentation: GPT-Image-1 and GPT-Image-2.

[5]Google, Build with Nano Banana 2; the public model name is Gemini 3.1 Flash Image.

Table C4: **Stage-1 Transfer Across Depth Backbones.** Each pair compares a frozen estimator with that estimator plus a trained injector. Full-frame composite depth is scored before RGB synthesis against targets from the same estimator; compare changes within each pair.

| Depth Backbone | Encoder | Method | In-Distribution | | Out-of-Distribution | |
|---|---|---|---|---|---|---|
| | | | AbsRel ↓ | $\delta_1$ ↑ | AbsRel ↓ | $\delta_1$ ↑ |
| Depth Anything V1 CVPR'24 (Yang et al., 2024a) | ViT-S | Baseline | 0.126 | 0.860 | 0.183 | 0.811 |
| | | **+ D2R** | **0.056** (−56%) | **0.972** | **0.100** (−45%) | **0.945** |
| | ViT-B | Baseline | 0.119 | 0.876 | 0.174 | 0.823 |
| | | **+ D2R** | **0.049** (−58%) | **0.978** | **0.088** (−49%) | **0.954** |
| | ViT-L | Baseline | 0.108 | 0.883 | 0.154 | 0.845 |
| | | **+ D2R** | **0.046** (−57%) | **0.980** | **0.083** (−46%) | **0.954** |
| Depth Anything V2 NeurIPS'24 (Yang et al., 2024b) | ViT-S | Baseline | 0.124 | 0.861 | 0.166 | 0.820 |
| | | **+ D2R** | **0.055** (−56%) | **0.971** | **0.097** (−42%) | **0.942** |
| | ViT-B | Baseline | 0.120 | 0.867 | 0.156 | 0.833 |
| | | **+ D2R** | **0.055** (−54%) | **0.973** | **0.080** (−49%) | **0.954** |
| | ViT-L | Baseline | 0.117 | 0.869 | 0.151 | 0.842 |
| | | **+ D2R** | **0.046** (−60%) | **0.977** | **0.075** (−50%) | **0.965** |
| Depth Anything V3 ICLR'26 (Lin et al., 2026) | ViT-S | Baseline | 0.146 | 0.824 | 0.179 | 0.782 |
| | | **+ D2R** | **0.104** (−29%) | **0.897** | **0.142** (−21%) | **0.851** |
| | ViT-B | Baseline | 0.140 | 0.838 | 0.160 | 0.811 |
| | | **+ D2R** | **0.093** (−34%) | **0.913** | **0.132** (−18%) | **0.863** |
| | ViT-L | Baseline | 0.114 | 0.861 | 0.153 | 0.811 |
| | | **+ D2R** | **0.073** (−36%) | **0.934** | **0.122** (−20%) | **0.889** |
| Depth Pro ICLR'25 (Bochkovskii et al., 2025) | ViT-L | Baseline | 0.080 | 0.910 | 0.161 | 0.859 |
| | | **+ D2R** | **0.053** (−34%) | **0.962** | **0.123** (−24%) | **0.935** |
| Metric-Anything ECCV'26 (Yang et al., 2026) | ViT-H/16+ | Baseline | 0.152 | 0.815 | 0.173 | 0.810 |
| | | **+ D2R** | **0.084** (−45%) | **0.929** | **0.094** (−46%) | **0.934** |
| InfiniDepth CVPR'26 (Yu et al., 2026) | ViT-L/16 | Baseline | 0.177 | 0.834 | 0.257 | 0.779 |
| | | **+ D2R** | **0.113** (−36%) | **0.924** | **0.159** (−38%) | **0.881** |

layouts. We adapt that strategy to reference-based compositing with FLUX.1-Fill and Redux, conditioning on reference and background depth. The evaluated baseline is this compositing-specific implementation; it does not use 3DIS's original LDM3D backbone or layout-conditioning adapter.

Table C5 compares this diffusion baseline with frozen Depth Anything V2 ViT-L (Yang et al., 2024b) and the same estimator with our width-640 injector. Both trained predictors use the same curated pairs and derive their supervision from the same paired composites and frozen depth estimator. The injector matches encoder features of the paired composite; the diffusion baseline denoises normalized $D = F(I)$ using the Stage-2 depth objective alone, without RGB or consistency losses. This tests direct depth-image generation against correction within the frozen estimator's feature space.

The diffusion route reuses the Stage-2 canvas layout, $1024 \times 1536$ resolution, and rank-256 LoRA with scaling 256 (App. C.3). Background and reference depths provide conditioning, and masked depth tokens are denoised into a composite-depth map over 28 sampling steps.

This is a Stage-1 depth-prediction comparison. We evaluate the FLUX route before adaptation and at its final checkpoint after 50K training iterations, matching the injector's iteration count. Full-frame scoring covers all 287 ID and 139 OOD cases before RGB synthesis. Predicted maps are min-max normalized to $[0, 1]$, stored as 8-bit PNGs, and scored after the same least-squares alignment as other depth predictions (App. C.4.1). The main paired tables score depth from RGB outputs.

Fine-tuning the FLUX route lowers ID AbsRel from 0.1262 to 0.1067 and OOD AbsRel from 0.1792 to 0.1740, reductions of 15.5% and 2.9%. The injector reaches 0.0462 and 0.0751, reducing error relative to the adapted FLUX route by 56.7% and 56.8%. Its advantage is therefore present on both familiar categories and categories absent from compositing training under this protocol.

These results favor feature correction over our diffusion-based composite-depth implementation on both splits. Architecture, conditioning, and objective change together, so this comparison evaluates the two depth-prediction strategies as implemented for compositing. The matched experiment in App. B.2.4 isolates the supervision target within the same injector, while Tab. C4 tests reuse across six depth backbones.

**Prediction Cost.** The injector corrects features within one feed-forward evaluation of the frozen depth estimator, whereas the tested FLUX route requires 28 denoising steps followed by VAE decoding into a depth image. The complete injector-equipped predictor takes 56.5 ms and 5.21 GB peak inference memory on RTX PRO 6000 (App. B.4). This avoids iterative sampling for composite-depth prediction. The FLUX timings in Tab. B4 measure Stage-2 RGB synthesis and are not used as timings for this depth-generation baseline.

This finding concerns the specified FLUX.1-Fill and Redux implementation. Marigold (Ke et al., 2024) shows that diffusion priors can support depth generalization, and Lotus (He et al., 2025) demonstrates the importance of the objective and sampling formulation for dense prediction. Our comparison therefore does not characterize diffusion-based depth prediction as a whole. Full-frame scores combine object and scene error without localizing the gain.

Table C5: Composite Depth Prediction with Feature Correction and Diffusion.

| Depth Predictor | Prediction | ID AbsRel ↓ | OOD AbsRel ↓ |
|---|---|---|---|
| Frozen Depth Anything V2 ViT-L NeurIPS'24 | Single Pass | 0.1168 | 0.1506 |
| Diffusion Depth (FLUX), No Fine-Tuning | 28 Steps | 0.1262 | 0.1792 |
| Diffusion Depth (FLUX), Fine-Tuned | 28 Steps | 0.1067 | 0.1740 |
| **Encoder Feature Injection (D2R; Ours)** | Single Pass | **0.0462** | **0.0751** |

## C.3 Rendering Backbones and Ablation Protocols

### C.3.1 Backbones and Prompt Conditioning

The renderer comparison tests whether the same composite-depth interface can support SD2-Inpainting (Rombach et al., 2022), SDXL-Inpainting (Podell et al., 2024), Qwen-Inpainting (Wu et al., 2025), and FLUX.1-Fill. Each uses the three-patch canvas, with depth observed throughout inference, a frozen autoencoder, and only LoRA trained. SD2 and SDXL predict noise; FLUX and Qwen predict velocity. The conditioning formats below are used consistently during training and inference; App. C.3.2 specifies training and sampling settings.

FLUX.1-Fill receives reference appearance through both the canvas and Redux. The Redux embedding occupies the text-conditioning stream, so this path constructs no text prompt. Any remaining text slots expected by the pipeline are filled with zeros to preserve its tensor layout.

SD2-Inpainting (Rombach et al., 2022), SDXL-Inpainting (Podell et al., 2024), and Qwen-Inpainting (Wu et al., 2025) receive reference appearance through the canvas and use text instructions without a separate image-prompt encoder. Each instruction has exactly two variable slots: the object noun {obj}, read from the sample's file name, and the insertion or replacement phrase, denoted {verb} in the compressed instruction or {action} in the full instruction. A single space serves as the negative prompt.

The CLIP text encoders of SD2-Inpainting and SDXL-Inpainting truncate at 77 tokens. Both therefore receive the compressed instruction, which contains 42 to 56 tokens under that tokenizer across the released splits: *"photorealistic composite: left, a scene with the {obj} {verb}, cast shadow, reflections, natural contact and lighting; top right, reference photo of the {obj}; bottom right, its depth map."* The {verb} slot is *inserted* for insertion and *replacing the original object* for replacement.

The Qwen-Inpainting text encoder accepts 512 tokens for sampling and 1024 for training. Its full instruction contains 115 to 152 tokens under the Qwen tokenizer: *"A composite image. Left half: a realistic scene where {action}. Top right: a clean reference photograph of a single {obj} on a plain background. Bottom right: a grayscale depth map matching the left half — bright pixels for nearby surfaces, dark pixels for distant surfaces — accurately reflecting the inserted {obj}'s 3D placement and contact geometry. Maintain photorealistic quality."*

For Qwen-Inpainting, the {action} slot is *"the {obj} from the top-right reference panel has been naturally inserted into the highlighted region — with cast shadows on the ground and adjacent surfaces, surface reflections and specular highlights, and physically plausible contact with the scene, all consistent with the existing lighting direction, materials, perspective, and color temperature"* for an insertion. For a replacement, it is *"the original object has been replaced with the {obj} from the top-right reference panel — preserving the original cast shadows, surface reflections, and physical contact patterns of the scene while adopting the visual identity of the {obj}"*.

The shorter instruction omits the full instruction's explicit depth-polarity description. Both retain guidance on shadows, reflections, contact, lighting, and the canvas layout. The instructions above are reproduced verbatim.

### C.3.2 Training and Sampling Settings

All trained Stage 2 ablation variants are evaluated at their final checkpoints after 100K training iterations. SD2-Inpainting, Qwen-Inpainting, and FLUX.1-Fill train rank-256 LoRA with scaling 256 using Prodigy (Mishchenko & Defazio, 2024) at a configured learning rate of 1.0. SDXL uses the AdamW settings given below. D2R renderers use exported Stage-1 predictions, the auxiliary objectives, and per-sample depth selection defined in App. B.3.

SD2-Inpainting (Rombach et al., 2022) uses a 512-pixel composite patch, a $512 \times 768$ canvas, batch size 12, and no gradient accumulation. White reference padding replaces the pipeline's edge-padding default. Sampling uses 28 steps, classifier-free guidance 7.0, and seed 0.

Qwen-Inpainting (Wu et al., 2025) uses a 768-pixel composite patch and a $768 \times 1152$ canvas to fit available training memory. Sampling uses 28 steps, true classifier-free guidance 4.0, and seed 42.

FLUX.1-Fill uses a 1024-pixel composite patch and a $1024 \times 1536$ canvas, with effective batch size 8 and four gradient-accumulation steps. Training takes approximately 35 h. FLUX-based D2R and matched Insert-Anything* use Redux, 28 sampling steps, guidance 30, and seed 0. This follows the 28-step Insert-Anything (Song et al., 2026) protocol; the FLUX.1-Fill example uses 50 steps.

SDXL-Inpainting (Podell et al., 2024) uses a 1024-pixel composite patch, a $1024 \times 1536$ canvas, batch size 2, four gradient-accumulation steps, fp32 adapter weights, and AdamW (Loshchilov & Hutter, 2019) at $5 \times 10^{-5}$. Both pretrained and adapted renderers sample with guidance 7.0 and 28 steps. Reference and depth patches remain fixed under Alg. 1. Appendix B.4 gives precision, latency, and training memory for the evaluated configurations.

To assess training stability, we train each stage five times under its default configuration with seeds 0, 1, 2, 3, and 4 and evaluate the final checkpoints. Relative changes across the applicable geometric, photometric, and identity metrics average approximately 1% overall. Sampling-seed checks separately indicated stable performance.

### C.3.3 Comparison Across Diffusion Backbones

Table 9 compares pretrained and D2R-adapted renderers on the same 139 OOD cases. Within each backbone, both receive identical Stage-1 predicted composite depth on the same canvas and use the sampling settings in App. C.3.2. Thus the pretrained comparator already receives depth, unlike the empty-depth Insert-Anything* baseline (Tab. 3). All four adapted renderers improve AbsRel and PSNR, supporting adaptation across the tested generators. Resolution, conditioning, and training differences prevent a controlled ranking across architectures.

### C.3.4 Loss Ablations and Sampling Comparisons

Table 7 evaluates the renderer through successive comparisons. Rows (a) and (b) test adding composite depth under RGB-only supervision. With depth supplied, rows (c) and (d) add depth denoising or RGB-depth consistency separately; comparing them with (f) tests their combination. Rows (e) and (f) retain both objectives and change only the consistency weighting from uniform to $1 - t$. All rows retain RGB supervision and matched data, backbone, LoRA, and training settings, using final checkpoints after 100K iterations. Conditioned rows (b) to (f) share per-sample 50:50 selection of

predicted or target depth during training, full-canvas corruption, and Stage-1 predicted depth at evaluation. Depth-patch denoising requires a depth prediction region and its target, whereas consistency supervises depth estimated from the generated RGB patch.

Table 8 instead changes only the training-time depth source. GT-only inputs, predicted-only inputs, and blending replace the default per-sample selection while retaining GT supervision and the other training settings. Each renderer is then evaluated with both target and predicted depth. Equation (B9) defines selection, and App. B.3.2 specifies all four inputs.

Latent Optimization (row (c) of Tab. 6) tests correction toward Stage-1 depth using released FLUX.1-Fill with Redux and no weight training. It applies the same correction at all 28 steps.

For this baseline, $\sigma_t$ denotes the sampler's noise level and $v_t = \mathrm{sg}(\epsilon_\theta(x_t, t, c))$ its detached velocity prediction. The clean estimate is $\hat{x}_0 = x_t - \sigma_t v_t$. We decode its composite patch, estimate depth with frozen Depth Anything V2 ViT-L (Yang et al., 2024b), and compare it with Stage-1 composite depth $\hat{D}$ using the normalization $A$ defined in App. B.3.2:

$$\begin{aligned} \mathcal{L}_{\text{latent}} &= \left\| A\Big(F\Big(\mathcal{D}(\hat{x}_0)\big|_{\Omega_{\text{comp}}}\Big)\Big) - \mathrm{sg}\Big(A(\hat{D})\Big) \right\|_1, \\ g_t &= \nabla_{x_t} \mathcal{L}_{\text{latent}}, \qquad x_t \leftarrow x_t - \gamma \sigma_t \|x_t\|_2 \frac{g_t}{\|g_t\|_2}, \qquad \gamma = 0.15. \end{aligned} \tag{C1}$$

For a nonzero gradient, the update normalizes its direction and scales the correction by the current latent norm and noise level. Thus $\gamma$ (`g_scale`) is a relative correction factor, not a conventional learning rate; the correction shrinks as $\sigma_t$ approaches zero. Detaching the velocity prediction restricts backpropagation to the RGB decoder and depth estimator, avoiding the denoiser's backward graph. The latent correction takes approximately 2 s per step on RTX PRO 6000.

Joint Denoising (row (d) of Tab. 6) trains the same backbone and canvas for 100K iterations but generates depth alongside RGB. Its conditioning depth is $F(I_{\text{bg}})$, while the targets are $D$ for depth and $I$ for RGB. The final checkpoint supplies the quantitative results in Tab. 6. Figure C1 visualizes this joint trajectory alongside D2R with fixed GT composite depth; the table evaluates D2R with Stage-1 predicted depth.

The geometry-provisioning strategies in Tab. 6 differ in both depth construction and training. Their results compare complete implementations rather than isolating conditioning time; the trajectory in Fig. C1 illustrates the conditioning schemes qualitatively.

## C.4 Evaluation Metrics

### C.4.1 Depth Alignment and Geometric Metrics

We evaluate the geometry of the generated RGB image through $D_{\text{out}} = F(\hat{I})$, using $D = F(I)$ as the paired target and frozen Depth Anything V2 ViT-L (Yang et al., 2024b) as $F$. Direct Stage-1 evaluation replaces $D_{\text{out}}$ with $\hat{D}$. Because relative-depth predictions can differ in scale and offset, we align the prediction to the target by ordinary least squares. The fitting support $\mathcal{B}$ consists of background pixels outside the target mask dilated with a $21 \times 21$ square kernel at the original evaluation resolution. If fewer than 100 background pixels remain, the fit uses the full frame:

$$D_{\text{aligned}} = sD_{\text{out}} + o, \qquad (s, o) = \arg\min_{a,b} \sum_{p \in \mathcal{B}} \big(aD_{\text{out},p} + b - D_p\big)^2. \tag{C2}$$

We solve the normal equations in closed form; degenerate fits use unit scale and a fitted offset. The aligned map is then scored over the full-frame pixel set $\mathcal{R}$:

$$\begin{aligned} \text{AbsRel} &= \frac{1}{|\mathcal{R}|} \sum_{p \in \mathcal{R}} \frac{\big|D_{\text{aligned},p} - D_p\big|}{\max\big(D_p,\, \varepsilon_{\text{d}}\big)}, \\ \text{RMSE} &= \sqrt{\frac{1}{|\mathcal{R}|} \sum_{p \in \mathcal{R}} \big(D_{\text{aligned},p} - D_p\big)^2}, \end{aligned} \tag{C3}$$

$$\delta_k = \frac{1}{|\mathcal{R}|} \sum_{p \in \mathcal{R}} \mathbf{1}\left[ \max\left( \frac{D^+_{\text{aligned},p}}{D^+_p},\, \frac{D^+_p}{D^+_{\text{aligned},p}} \right) < \tau^k \right], \tag{C4}$$

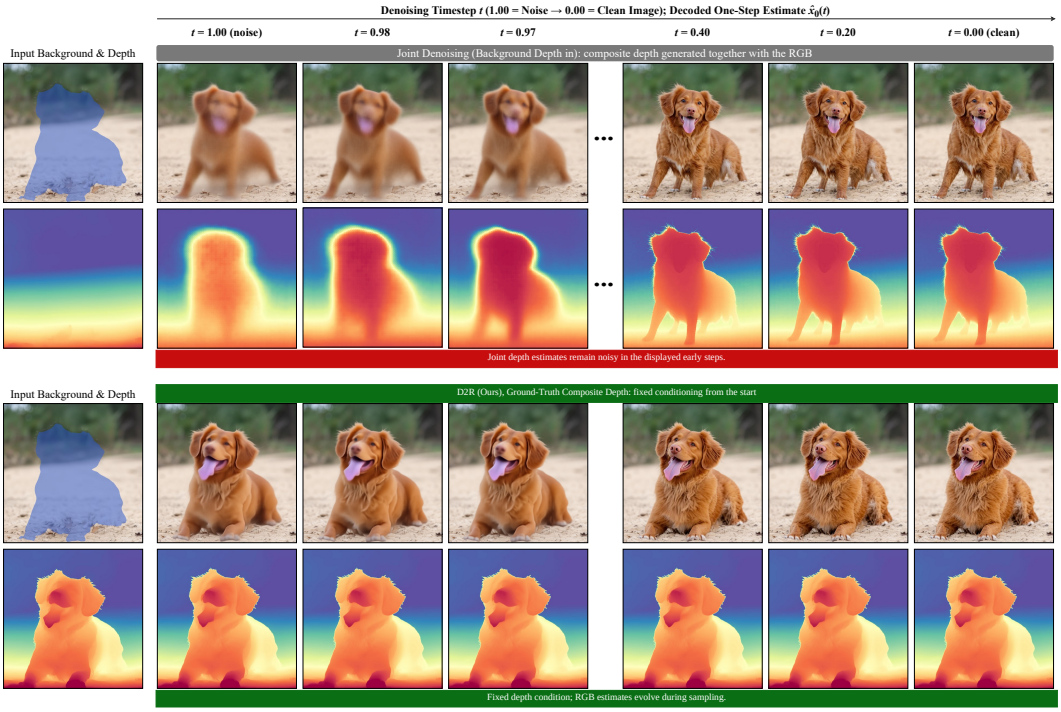


Figure C1: **Denoising Trajectories with Joint Prediction and Fixed Ground-Truth Depth.** The upper branch predicts depth jointly with RGB; the lower D2R branch uses GT composite depth as a fixed condition. Tiles show one-step estimates from one example. Table 6 instead evaluates D2R using composite depth predicted by Stage 1.

Here $\varepsilon_{\mathrm{d}} = 0.05$, $D_p^+ = \max(D_p, \varepsilon_{\mathrm{d}})$, and $D_{\mathrm{aligned},p}^+ = \max(D_{\mathrm{aligned},p}, \varepsilon_{\mathrm{d}})$. AbsRel floors only its denominator; its numerator and RMSE use the original aligned and target values. For ratio accuracy, both maps are floored before taking ratios, and every pixel remains in the average. We use $\tau = 1.25$ (Yang et al., 2024a;b) and report $\delta_1$. These metrics compare aligned estimator outputs rather than calibrated physical depth.

### C.4.2 Paired Uncertainty and Regional Evaluation

We test whether D2R's average geometric improvement persists across cases, rather than depending on a few large gains. D2R uses Stage-1 predicted depth in all comparisons; the rendered-image cases are the same as in the main paired evaluation. For each case, $\Delta = \mathrm{AbsRel}_{\mathrm{D2R}} - \mathrm{AbsRel}_{\mathrm{baseline}}$, so negative differences favor D2R. We compute percentile intervals from 10,000 paired bootstrap resamples with seed 0, sampling individual cases while retaining the paired method scores.

Insertion and replacement cases can share an image ID, but the primary bootstrap samples them separately. For the OOD comparison with Insert-Anything* (Song et al., 2026), a cluster bootstrap over 82 image IDs gives $[-0.0785, -0.0286]$, while the bias-corrected and accelerated (BCa) interval is $[-0.0749, -0.0354]$. Both exclude zero, as does the case-level percentile interval in Tab. C6. The improvement persists under both alternative bootstrap procedures.

Table C6 compares D2R with matched Insert-Anything* and MimicBrush (Chen et al., 2024a). Wins exclude differences within the AbsRel tie tolerance $|\Delta| \leq 0.001$. To measure sensitivity to the largest gains, we remove the $\mathrm{round}(0.1n)$ cases most favorable to D2R: 14 OOD or 29 ID cases. The remaining mean difference is negative in all four comparisons. Large improvements contribute to the aggregate gain, but they do not account for the entire full-frame advantage.

To determine where the improvement occurs, we also score the object interior, its boundary, and the background. Predicted depth is resized to the original target-depth resolution before a $21 \times 21$ square kernel is applied to the target mask $M$. The interior is $\mathrm{erode}(M)$; the boundary band is $\mathrm{dilate}(M) \setminus \mathrm{erode}(M)$, extending approximately ten pixels to each side of the boundary; and the background is the complement of the dilation. All regions reuse the alignment in Eq. (C2), with no

Table C6: Paired Full-Frame AbsRel Differences and Sensitivity to Large Gains.

| Comparator | Split | Mean $\Delta$ | 95% CI | Wins / $n$ | Trimmed $\Delta$ |
|---|---|---|---|---|---|
| Insert-Anything* AAAI'26 | OOD | $-0.0514$ | $[-0.0717, -0.0337]$ | 103 / 139 | $-0.0218$ |
| Insert-Anything* AAAI'26 | ID | $-0.0147$ | $[-0.0198, -0.0101]$ | 197 / 287 | $-0.0054$ |
| MimicBrush NeurIPS'24 | OOD | $-0.0231$ | $[-0.0317, -0.0148]$ | 95 / 139 | $-0.0103$ |
| MimicBrush NeurIPS'24 | ID | $-0.0187$ | $[-0.0233, -0.0141]$ | 201 / 287 | $-0.0097$ |

additional regional fit. Empty regions are omitted from that region's mean, and the valid case count is retained. Background scores measure residual disagreement on the alignment support.

On the 139 OOD cases, D2R lowers interior and boundary-band AbsRel relative to both comparators (Tab. C7). D2R has interior AbsRel of 0.0509 and boundary-band AbsRel of 0.1491. All four paired intervals exclude zero, showing that the full-frame advantage also extends to the inserted object and nearby scene content. These differences use unrounded case scores. The result is metric-specific: the OOD interior $\delta_1$ gain over Insert-Anything* reverses after excluding the largest 10% of improvements. The AbsRel evidence therefore supports regional gains without implying the same stability for every metric.

Table C7: Object and Boundary AbsRel on OOD Cases.

| Comparator | Region | Baseline AbsRel | Mean $\Delta$ | 95% CI |
|---|---|---|---|---|
| Insert-Anything* AAAI'26 | Interior | 0.0734 | $-0.0225$ | $[-0.0318, -0.0138]$ |
| | Band | 0.1887 | $-0.0395$ | $[-0.0560, -0.0230]$ |
| MimicBrush NeurIPS'24 | Interior | 0.1119 | $-0.0610$ | $[-0.0747, -0.0485]$ |
| | Band | 0.1888 | $-0.0396$ | $[-0.0562, -0.0225]$ |

### C.4.3 Cross-Scorer Evaluation

We hold the generated images fixed and change the depth estimator used to score them. Depth Anything V1 (Yang et al., 2024a), V2 (Yang et al., 2024b), V3 (Lin et al., 2026), Depth Pro (Bochkovskii et al., 2025), and Metric-Anything (Yang et al., 2026) each evaluate all sixteen methods on the same 287 ID and 139 OOD cases. Each scorer estimates depth from both the generated and ground-truth images, then applies Eq. (C2) and full-frame scoring. D2R retains its adopted Stage-1 predictions throughout. Depth Pro and Metric-Anything provide no training targets for this D2R configuration; their trained injector variants belong to the separate Stage-1 ablations.

This comparison uses floating-point depth maps, whereas the main protocol stores 8-bit PNGs, so its values are reported separately. Tables C8 and C9 give AbsRel, ranks, and Spearman correlations relative to the Depth Anything V2 ordering; DAv1 to DAv3 abbreviate the corresponding Depth Anything versions. Absolute errors should be compared within each scorer, because their depth representations and scales differ.

D2R ranks first under every scorer on both splits, although the remaining methods change order. Across the four alternative scorers, Spearman correlations with the Depth Anything V2 ordering range from 0.888 to 0.985 on ID and from 0.824 to 0.976 on OOD. The leading result is therefore stable even where the broader ranking varies, supporting the geometric improvement beyond the estimator that supplied training targets.

### C.4.4 Photometric and Identity Metrics

For paired photometric evaluation, PSNR and LPIPS (Zhang et al., 2018b) compare the generated composite $\hat{I}$ with its target $I$ after resizing both to the 512-pixel evaluation resolution.

Paired and unpaired identity evaluation share the same image encoders: IsoCLIP (Magistri et al., 2026) for CLIP-I and DINOv3 ViT-L/16 (Siméoni et al., 2025) for DINO-I. IsoCLIP uses the SigLIP2 (Tschannen et al., 2025) backbone `ViT-gopt-16-SigLIP2-384` with `webli` weights. The call `IsoCLIP(model, weights, 200, 50)` removes the top 200 and bottom 50 SVD directions from the image and text projections.

Table C8: Cross-Scorer AbsRel and Ranks on the In-Distribution Split.

| Method | DAv2 | | DAv1 | | DAv3 | | Depth Pro | | Metric-Anything | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank |
| **D2R (Ours)** | **0.048** | **1** | **0.049** | **1** | **0.057** | **1** | **0.040** | **1** | **0.090** | **1** |
| Insert-Anything* AAAI'26 (Song et al., 2026) | 0.064 | 2 | 0.066 | 3 | 0.074 | 3 | 0.051 | 2 | 0.099 | 2 |
| MimicBrush NeurIPS'24 (Chen et al., 2024a) | 0.067 | 3 | 0.064 | 2 | 0.081 | 4 | 0.053 | 3 | 0.134 | 4 |
| UniCombine ICCV'25 (Wang et al., 2025) | 0.073 | 4 | 0.073 | 4 | 0.070 | 2 | 0.058 | 4 | 0.143 | 6 |
| DIRECT ICML'26 (Gong et al., 2026b) | 0.089 | 5 | 0.083 | 5 | 0.089 | 5 | 0.059 | 5 | 0.124 | 3 |
| BIFRÖST NeurIPS'24 (Li et al., 2024a) | 0.099 | 6 | 0.096 | 6 | 0.098 | 8 | 0.064 | 7 | 0.150 | 7 |
| HiFi-Inpaint CVPR'26 (Liu et al., 2026) | 0.105 | 7 | 0.098 | 7 | 0.091 | 6 | 0.062 | 6 | 0.140 | 5 |
| Paint-by-Example CVPR'23 (Yang et al., 2023) | 0.113 | 8 | 0.102 | 8 | 0.105 | 9 | 0.070 | 9 | 0.198 | 12 |
| DreamFuse ICCV'25 (Huang et al., 2025) | 0.129 | 9 | 0.124 | 9 | 0.098 | 7 | 0.070 | 8 | 0.157 | 8 |
| SHINE ICLR'26 (Lu et al., 2026) | 0.164 | 10 | 0.152 | 10 | 0.125 | 11 | 0.088 | 10 | 0.171 | 10 |
| AnyDoor CVPR'24 (Chen et al., 2024b) | 0.176 | 11 | 0.162 | 12 | 0.143 | 14 | 0.097 | 12 | 0.198 | 13 |
| Personalize-Anything AAAI'26 (Feng et al., 2026) | 0.177 | 12 | 0.162 | 13 | 0.130 | 12 | 0.111 | 14 | 0.203 | 15 |
| Nano Banana 2 Tested 2026-08 | 0.179 | 13 | 0.152 | 11 | 0.117 | 10 | 0.095 | 11 | 0.163 | 9 |
| TF-ICON ICCV'23 (Lu et al., 2023) | 0.207 | 14 | 0.198 | 15 | 0.155 | 15 | 0.110 | 13 | 0.202 | 14 |
| GPT-Image-2 Tested 2026-08 | 0.209 | 15 | 0.197 | 14 | 0.136 | 13 | 0.112 | 15 | 0.188 | 11 |
| GPT-Image-1 Tested 2026-08 | 0.470 | 16 | 0.509 | 16 | 0.291 | 16 | 0.215 | 16 | 0.358 | 16 |
| Spearman $\rho$ vs. DAv2 | 1.000 | | 0.985 | | 0.941 | | 0.979 | | 0.888 | |

DINO-I uses the pretrained `facebook/dinov3-vitl16-pretrain-lvd1689m` checkpoint with $512 \times 512$ inputs and its [CLS] descriptor. For both metrics, we L2-normalize the image descriptors, compute the cosine similarity for each image pair, and average over cases.

The comparison target depends on whether a paired composite is available. Paired identity uses GT and predicted composite crops from the same region defined by the GT object mask, without replacing the surrounding pixels with white. Every method uses the same 287 ID and 139 OOD cases after the common exclusions. Unpaired identity compares the reference object with the predicted composite using the crops in App. C.1.3. Because both the target and crop convention change, the scores are not directly comparable across protocols. Identity and paired geometry metrics serve complementary roles: embedding agreement measures similarity to the object, while depth agreement evaluates the result in the target scene.

### C.5 Limitations

**Errors in the Geometric Condition.** Fixed depth can pass an incorrect footprint, contact height, or occlusion order into RGB. Viewing predicted depth beside the generated image helps distinguish the supplied geometry from the renderer's response. Target-depth substitution (Tabs. 2 and 3) measures the benefit of a better condition without assigning every remaining error to one stage. Multiple geometries may fit the inputs, whereas paired evaluation uses one target. Relative depth leaves metric scale undetermined, and thin or transparent structures remain difficult.

**Color and Texture Harmonization.** A geometrically plausible insertion can still mismatch the scene's appearance. The cat in Fig. D4 (row four) fits the target region but retains photographic fur shading in an ink drawing; the church in Fig. D3 (row seven) retains color in a grayscale scene. D2R can reflect illustrated styles, yet larger palette and texture changes remain difficult. Depth specifies structure rather than surface appearance, so it cannot alone resolve RGB-to-grayscale harmonization. Photographic training may also favor reference retention, although its contribution is not isolated here. These limitations motivate geometry-guided harmonization and relighting as future work.

Table C9: Cross-Scorer AbsRel and Ranks on the Out-of-Distribution Split.

| Method | DAv2 | | DAv1 | | DAv3 | | Depth Pro | | Metric-Anything | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank | AbsRel ↓ | Rank |
| **D2R (Ours)** | **0.068** | **1** | **0.072** | **1** | **0.105** | **1** | **0.103** | **1** | **0.077** | **1** |
| MimicBrush NeurIPS'24 (Chen et al., 2024a) | 0.091 | 2 | 0.132 | 2 | 0.130 | 2 | 0.137 | 3 | 0.160 | 4 |
| Insert-Anything* AAAI'26 (Song et al., 2026) | 0.120 | 3 | 0.136 | 3 | 0.131 | 3 | 0.139 | 5 | 0.151 | 2 |
| BIFRÖST NeurIPS'24 (Li et al., 2024a) | 0.121 | 4 | 0.138 | 4 | 0.143 | 6 | 0.138 | 4 | 0.166 | 5 |
| HiFi-Inpaint CVPR'26 (Liu et al., 2026) | 0.133 | 5 | 0.157 | 7 | 0.133 | 4 | 0.145 | 9 | 0.156 | 3 |
| Paint-by-Example CVPR'23 (Yang et al., 2023) | 0.137 | 6 | 0.146 | 6 | 0.156 | 9 | 0.169 | 10 | 0.216 | 10 |
| DIRECT ICML'26 (Gong et al., 2026b) | 0.138 | 7 | 0.146 | 5 | 0.135 | 5 | 0.126 | 2 | 0.169 | 6 |
| Personalize-Anything AAAI'26 (Feng et al., 2026) | 0.189 | 8 | 0.201 | 9 | 0.154 | 8 | 0.171 | 11 | 0.213 | 9 |
| UniCombine ICCV'25 (Wang et al., 2025) | 0.192 | 9 | 0.198 | 8 | 0.150 | 7 | 0.139 | 7 | 0.229 | 11 |
| SHINE ICLR'26 (Lu et al., 2026) | 0.220 | 10 | 0.238 | 11 | 0.169 | 12 | 0.172 | 12 | 0.209 | 8 |
| DreamFuse ICCV'25 (Huang et al., 2025) | 0.228 | 11 | 0.241 | 12 | 0.162 | 11 | 0.139 | 6 | 0.233 | 12 |
| Nano Banana 2 Tested 2026-08 | 0.238 | 12 | 0.211 | 10 | 0.160 | 10 | 0.144 | 8 | 0.207 | 7 |
| AnyDoor CVPR'24 (Chen et al., 2024b) | 0.260 | 13 | 0.261 | 13 | 0.187 | 13 | 0.189 | 13 | 0.274 | 14 |
| GPT-Image-2 Tested 2026-08 | 0.302 | 14 | 0.303 | 14 | 0.196 | 14 | 0.199 | 14 | 0.270 | 13 |
| TF-ICON ICCV'23 (Lu et al., 2023) | 0.360 | 15 | 0.400 | 15 | 0.240 | 15 | 0.239 | 15 | 0.342 | 15 |
| GPT-Image-1 Tested 2026-08 | 0.492 | 16 | 0.549 | 16 | 0.301 | 16 | 0.296 | 16 | 0.410 | 16 |
| Spearman $\rho$ vs. DAv2 | 1.000 | | 0.976 | | 0.956 | | 0.824 | | 0.906 | |

**Scope of the Evidence.** Agreement across depth scorers supports the ranking without establishing physical accuracy or local contact correctness. Regional evaluation (App. C.4.2) addresses errors hidden by full-frame averages. The matched RGB baseline controls data and renderer but cannot separate the pretrained depth prior from the added 272.4M injector parameters. Holding both fixed, the feature-versus-depth comparison isolates supervision (App. B.2.4).

## D ADDITIONAL QUALITATIVE RESULTS

All displayed D2R examples use Stage-1 predicted depth. Figures D1–D5 follow Fig. 1: columns seven and eight show the predicted depth and resulting RGB. The OOD examples contain object tags absent from compositing training.

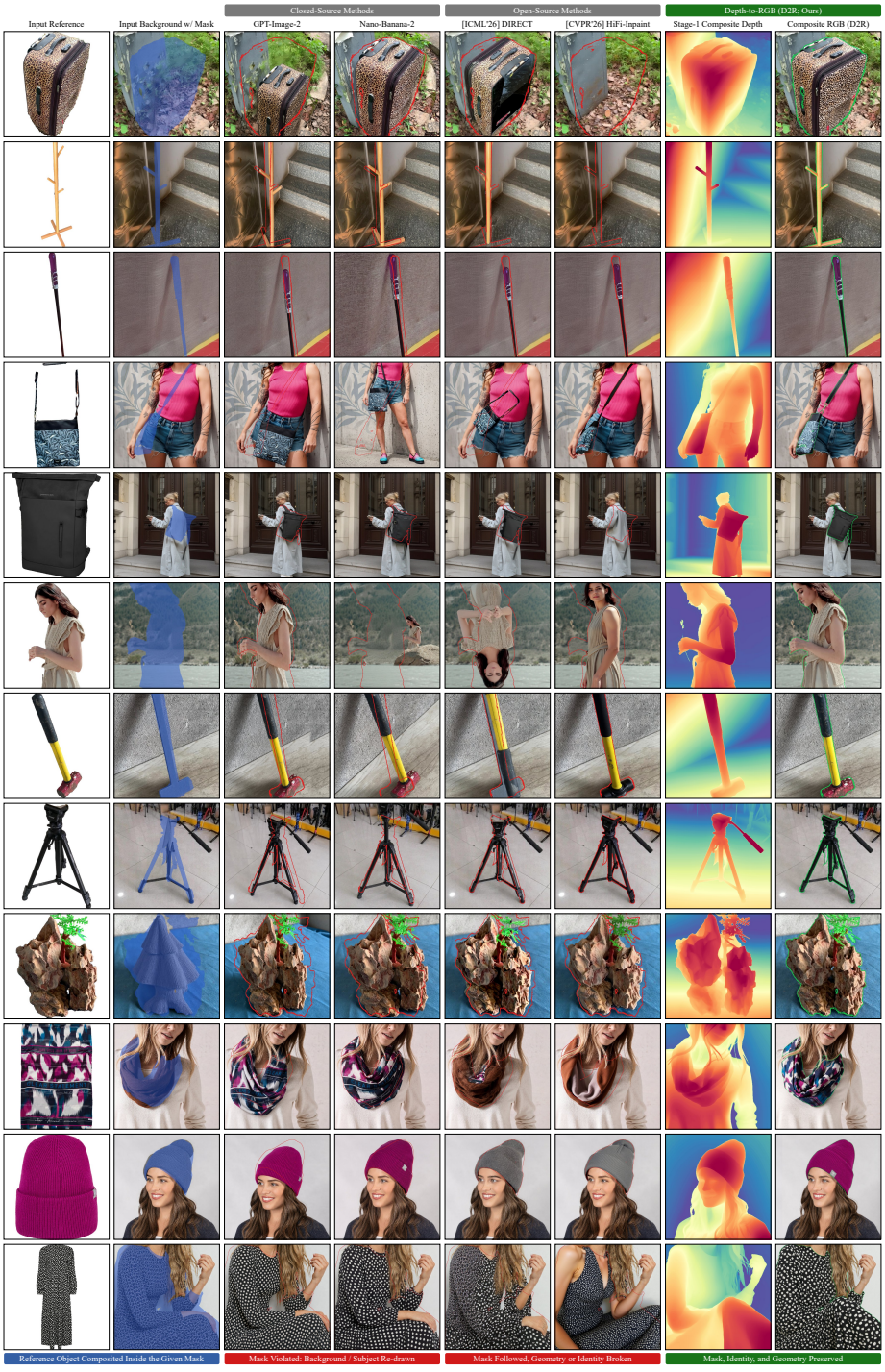


Figure D1: Qualitative Results on the AnyInsertion++ In-Distribution (ID) Split.




Figure D2: Qualitative Results on the AnyInsertion++ Out-of-Distribution (OOD) Split.

**Depth in Illustrated Scenes.** The bottom three rows of Fig. D3 depict cartoon breakfast-table, living-room, and forest scenes; the first four rows of Fig. D4 show illustrated landscapes and an ink drawing. Their depth and RGB columns show how predicted shape and placement translate into the composite. Appendix C.5 discusses the remaining harmonization errors.

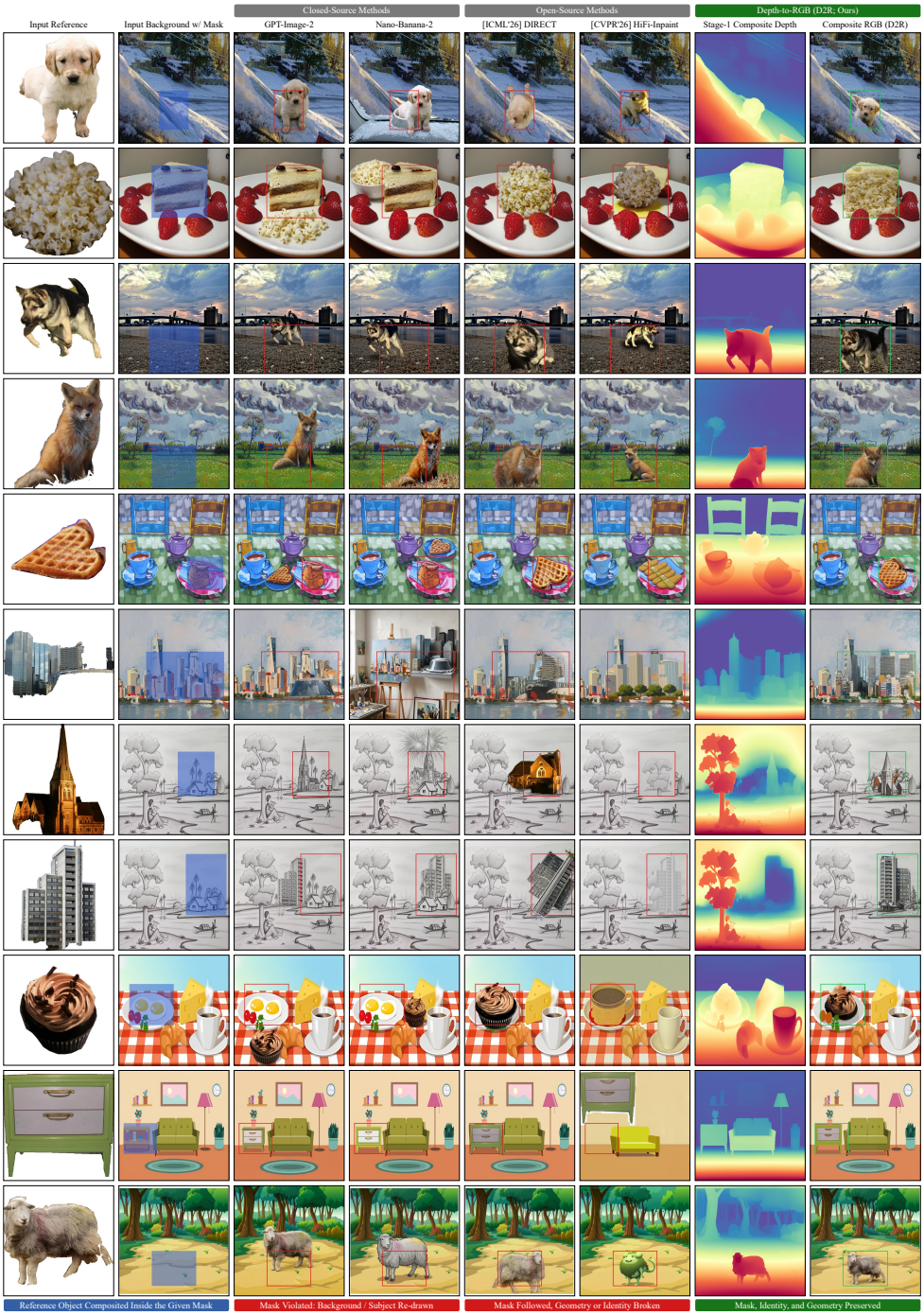


Figure D3: Qualitative Results on TF-ICON (Lu et al., 2023).

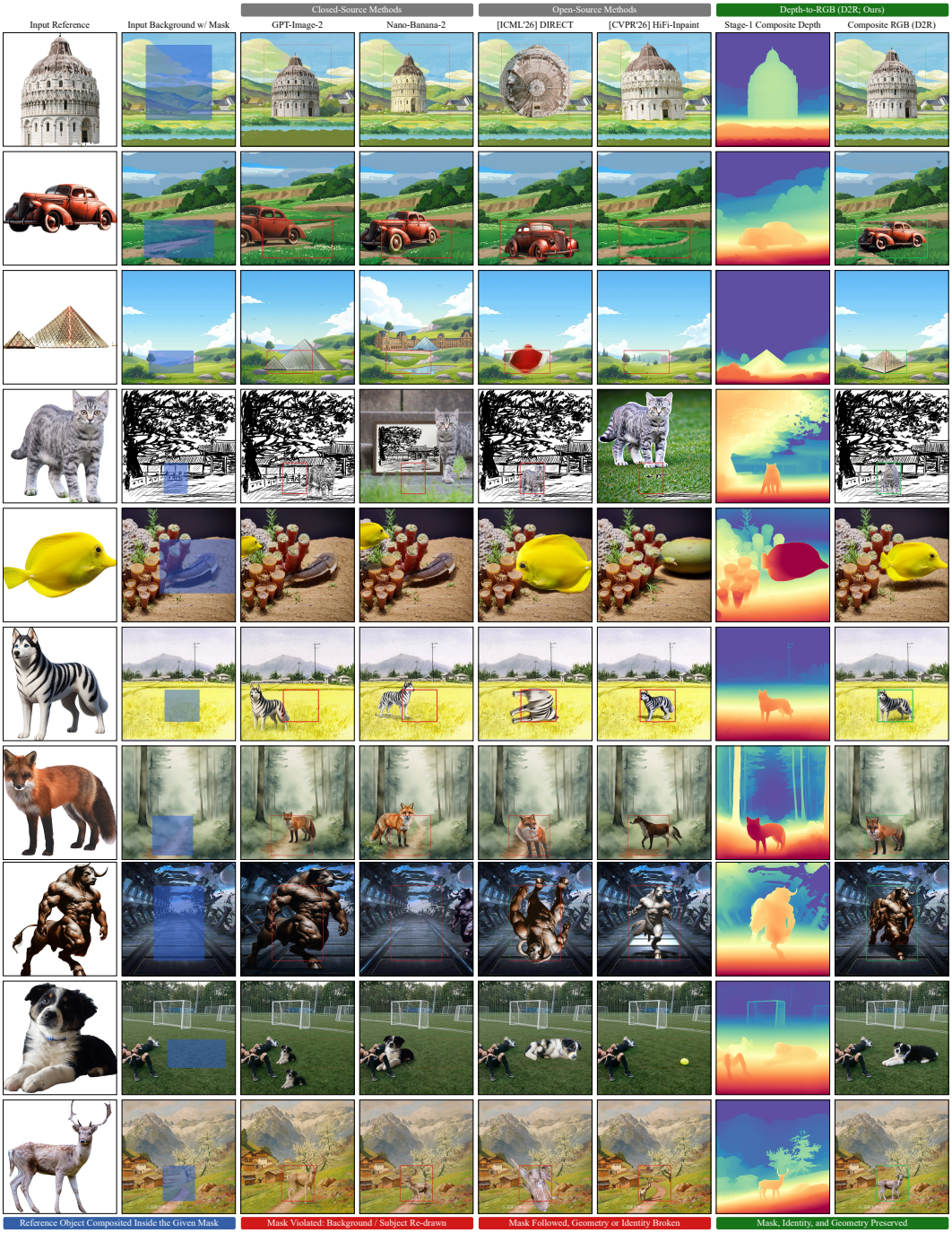


Figure D4: Qualitative Results on AIComposer (Li et al., 2025).

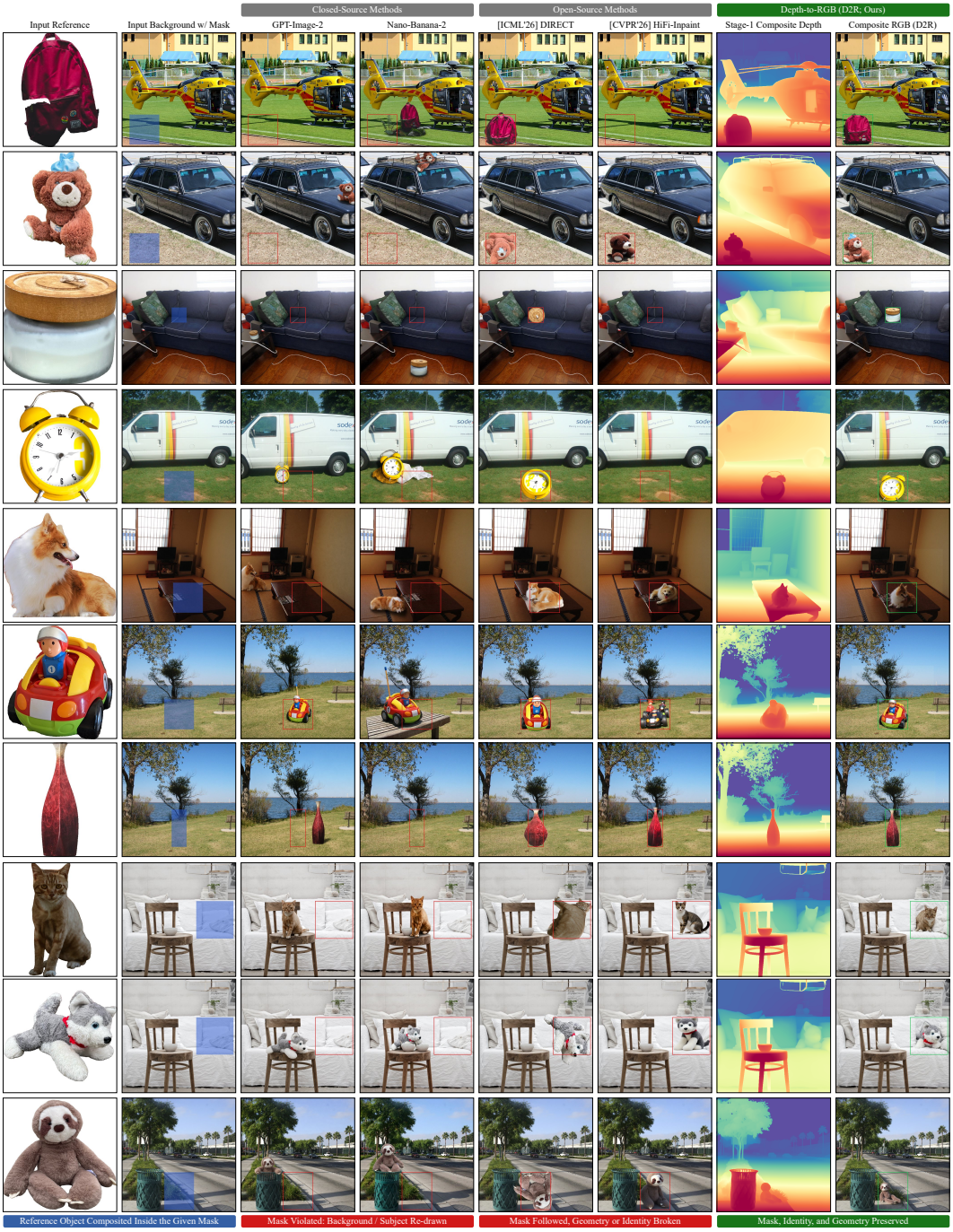


Figure D5: Qualitative Results on ComplexCompo (Lu et al., 2026).

**Point-Cloud Views.** Figure D6 shows each output back-projected with its own estimated depth and the background alignment in Eq. (C2). Off-axis views expose inconsistencies in these monocular predictions; they do not independently measure 3D geometry. Points crossing depth discontinuities are removed, and pale regions are disocclusions unsupported by the input view. The frustum marks the input camera. Dotted circles mark the largest inserted-object region with mean normalized disparity above $0.085$ relative to target depth, using the same rule for every method. No displayed D2R panel exceeds this criterion.




Figure D6: Composites Viewed as Point Clouds.