DiTailed: Ensuring Visual Object Consistency in Text-Image-to-Image Flow Matching Models

Francesco Taioli[1], Daniel Coelho[1], Iaroslav Melekhov[1], Roberto Alcover-Couso[1], Jose Miguel Grande Saiz[1], Virginia Fernandez Arguedas[1], and Artur Bekasov[2,⋆]

[1] Amazon, {ftaioli,dfcoelho,imlkhv,ralcover,jgrande,virfer}@amazon.com
[2] Faculty, artur.bekasov@faculty.ai

Fig. 1: ***ABO-Edit*** benchmarks *visual consistency*: the degree to which editing models preserve an object's visual attributes, *e.g.*, texture, color, geometry, and fine-grained details. Spanning 12K+ samples over diverse objects and edits, it systematically probes whether these attributes survive changes in viewpoint, context, and appearance.

Abstract. Despite remarkable progress in text-guided image editing, generative models frequently fail to preserve visual object consistency, defined as the preservation of a subject's key attributes throughout the editing process. We address this limitation through three contributions. First, we introduce ABO-Edit, a dataset specifically designed to study object consistency, comprising over 12,000 triplets of source images, editing prompts, and high-quality target images rendered from artist-designed 3D assets, with multi-view coverage and human-verified quality control. Second, we uncover an overlooked property of image-editing rectified flow models: the conditioning embedding space, not directly supervised during training, encodes a prediction of the final generated image even at high noise levels. Third, exploiting this finding, we propose FlowMirror, a parameter-free auxiliary loss that supervises this conditioning embedding space. Without architectural changes, our method improves generation quality across several metrics over baselines. Page: francescotaioli.github.io/DiTailed

Keywords: Rectified Flow Models · Visual Object Consistency

⋆ Work done while at Amazon.

Fig. 2: *Visual object inconsistencies* observed in current image-editing models: (*left*) pillow patterns are not preserved, (*center*) cup holder and rivets are missing or poorly reconstructed, and (*right*) sofa geometry is entirely altered with inconsistent texture.

1 Introduction

The advent of large-scale generative models [5, 16, 17, 26, 36, 52] has revolutionized the field of image synthesis, enabling the creation of high-fidelity images from diverse input modalities. These capabilities could potentially show value across various domains, including creative design [50], digital media production, and content creation [28].

Despite these significant advances, several challenges remain unresolved. Even with seemingly straightforward prompts specifying background removal or basic geometric transformations (*e.g.*, rotation), state-of-the-art generative models frequently fail to maintain *visual object consistency* [10, 18, 55, 57], defined as the preservation of an object's intrinsic visual attributes, including geometry, texture, color, and fine-grained details, throughout the image generation process, except where explicitly requested by the conditioning prompt.

We illustrate such failure cases in Fig. 2, where current large-scale open-weights image generation foundation models can struggle to keep the object consistent. As shown, when prompted to rotate an object and remove the background, the model fails to preserve critical visual details: the pattern on the pillow is altered, the support rings and cup-holders are missing or poorly generated, and the sofa's shape is substantially distorted with a texture that misrepresents the original product. This lack of fine-grained element preservation may limit the applicability of current generative models for downstream applications that could demand pixel-level visual fidelity, such as product visualization, 3D digital content creation [55], novel view synthesis [57], and product standardization [15].

To address these limitations, we make three contributions: a dedicated benchmark for measuring visual object consistency, novel analytical insights into the internal behavior of rectified flow editing models, and a parameter-free auxiliary loss that leverages these insights to improve generation quality.

First, we introduce *ABO-Edit*, a curated benchmark for visual object consistency built on the Amazon Berkeley Objects (ABO) dataset [11][3]. As shown in Fig. 1, ABO-Edit contains triplets consisting of: *(i)* a high-definition source image; *(ii)* a textual prompt specifying the intended transformation, and *(iii)* a

[3] Amazon Berkeley Objects (ABO) dataset is publicly available under CC BY 4.0 License at https://amazon-berkeley-objects.s3.amazonaws.com/.

high-definition target image representing the desired output. Source images depict products in "lifestyle" settings, *i.e.*, real-world scenes with complex backgrounds, occlusions, and contextual clutter, posing a particularly challenging scenario for generative models that must disentangle object-specific features from high visual noise. ABO-Edit covers a diverse set of product types, including furniture (*e.g.*, sofas, chairs), wall art, and decorative items. To ensure ground-truth object consistency, target images are rendered in high resolution (1024×1024) from the corresponding artist-designed 3D models using Blender [6], guaranteeing accurate preservation of object geometry, texture, and fine-grained details. We render target images on a white background, with realistic shadows. Furthermore, our benchmark includes multi-view rendering of each object (*i.e.*, *left*, *front* and *right* views) at different rotation angles, which allows evaluating how well generative models maintain object consistency under object orientation transformations.

Second, we provide novel insights into the behavior of state-of-the-art image-editing models [1, 5, 26, 52][4] during the editing process. In particular, we identify that a specific model family already encodes a latent prediction of the final output in the conditioning embedding space of the input image, even at high-noise levels. Remarkably, if supervised, this latent prediction exhibits higher similarity to the target than the velocity estimate at early denoising steps, providing insight into the effective number of inference steps required for convergence.

Building on these findings, we propose a parameter-free auxiliary loss that exploits this intrinsic predictive behavior to enhance object consistency during generation, showing improvements in generation quality across multiple metrics and architectures. In summary, this paper makes the following contributions:

i. We introduce *ABO-Edit*, a benchmark for training and evaluating object consistency in image editing, featuring $12k+$ high-quality human-reviewed source-prompt-target triplets with ground-truth 3D renderings and degree-level orientation annotations. The dataset will be released upon acceptance;
ii. We identify that rectified flow editing models encode target predictions in the conditioning embedding space even at high noise levels; and
iii. We propose *FlowMirror*, a parameter-free auxiliary loss that supervises this embedding space to improve generation quality across metrics over baselines.

2 Related Works

This section reviews generative models; related datasets in Sec. 7.1.6 (*Appendix*).
Diffusion Models. Diffusion models [21, 44, 54] have emerged as a powerful class of generative models, achieving state-of-the-art results in modeling high-dimensional data distributions. While early efforts demonstrated strong performance, their widespread adoption was enabled by the computational efficiency gained through operating in a compressed latent space [42]. Subsequent advances in classifier-free guidance [22], deterministic sampling [45], and architectural design [25] have further enhanced both sample quality and practical deployment.

[4] All evaluated models are open-weight and publicly available.

These models have revolutionized generative modeling with applications spanning image synthesis [21, 36, 41, 42], video generation [20, 23], and 3D generation [37].
Rectified Flow Models. Rectified flow models [29, 31] simplify the generative process by learning straight trajectories between noise and data distributions, enabling more efficient sampling with fewer evaluation steps through linear probability paths. Concurrently, architectural advances have shifted from U-Net-based architectures to transformer-based designs. The Diffusion Transformer (DiT) [35] demonstrated that the classic transformer architecture can effectively replace the standard U-Net architecture, showing promising scaling properties. Furthermore, the MultiModal Diffusion Transformer (MM-DiT) [14] extended this approach to jointly model multiple modalities, *e.g.*, with images and text. Modern rectified flow models leverage powerful pretrained text encoders for conditioning. SD3 [14] and Flux [26] employ a combination of CLIP [39] and T5 [40] encoders to capture both semantic alignment and fine-grained textual understanding, while Qwen-Image [52] obtains semantic embeddings from a VLM [4]. Building on these foundations, recent works have demonstrated the effectiveness of large-scale, rectified flow models for video and image synthesis [5, 9, 16, 17, 26, 36, 46, 47, 52, 60], establishing new benchmarks for visual generation quality.

3 Dataset

We introduce ABO-Edit, a curated dataset for training and evaluating generative models on *Visual Object Consistency*. ABO-Edit addresses the challenging task of transforming *"Lifestyle"* images (depicting products in complex real-world usage scenarios) into *studio-quality* representations: the same product isolated on a white background with realistic shadow, rotated and tilted to a precisely specified angle (see Fig. 1). This task addresses a general challenge in e-commerce platforms [13, 15], where standardized and high-fidelity product imagery is commonly desired. Prior research in e-commerce has shown that *inconsistent* product imagery may affect user experience and search effectiveness [8, 32]. Specifically, each sample comprises a triplet $\langle x_{\text{src}}, p_{\text{src}\to\text{trg}}, x_{\text{trg}} \rangle$, where x_{src} denotes a source lifestyle image, $p_{\text{src}\to\text{trg}}$ represents a detailed editing prompt including fine-grained rotation angles, and x_{trg} is the corresponding ground-truth target image rendered from a 3D asset. Beyond the editing task evaluated here, ABO-Edit supports the reverse task (*studio-to-lifestyle* generation), single-image 3D reconstruction, and text-to-3D generation.

3.1 Dataset Construction

Our dataset is built upon the Amazon Berkeley Objects (ABO) Dataset [11], a large-scale, publicly available collection of high-quality images depicting household objects. ABO provides, for more than 7,900 products, high-quality, artist-created 3D models with 4K texture maps and physically-based materials. In the following, we describe how we construct ABO-Edit.

3.2 Filtering procedure

We implement a two-stage automatic filtering pipeline, described below.
Product-level filtering. To ensure high-quality $\langle x_{\text{src}}, x_{\text{trg}} \rangle$ pairs, we implement an automatic filtering pipeline, described as follows. We first exclude: *(i)* product types with minimal transformation requirements (*e.g.*, phone cases); *(ii)* product types not associated with 3D models (ABO [11] provides 3D models for more than 7,900 products); and *(iii)* product types where rendering is challenging due to material properties (*e.g.*, mirrors, transparent objects).
Image-level filtering. We then discard source images x_{src} lacking "*lifestyle*" context. To automate this process, we curate a small annotated dataset and train an MLP classifier on CLIP [39] embeddings to distinguish lifestyle images from studio-shot or plain product images. The classifier achieves 90% precision and 82% recall. Images passing this filter are retained as source images x_{src}.

3.3 Target Image Generation

To create corresponding target images x_{trg}, we leverage the 3D models associated with each product sample. Specifically, using the Blender Python API[5], we render high-quality RGB images of the products at 1024×1024 resolution.

Notably, for each source image x_{src}, we generate three target images x_{trg} corresponding to *front*, *left*, and *right* views of the object. For each view, we define a valid range of camera angles and randomly sample both an azimuth angle and an elevation angle relative to the object's front-facing coordinate system. The camera is then positioned accordingly, ensuring the object remains within the field of view. We show samples in the *Appendix*, Fig. 9.

3.4 Prompt Generation

We now describe the generation of the prompt $p_{\text{src}\to\text{trg}}$, which specifies the desired transformation from source to target image. We begin with a static template, shared across all samples, which we populate with sample-specific information. The full template is provided in the *Appendix* (Sec. 7.1.1).

Specifically, we populate the template with two sample-specific fields: *(i) Object category and description*: we employ a VLM to generate a detailed description of the object and extract its category from x_{trg} (the product is rendered on a white background). Including a fine-grained description enables the editing model to disambiguate the target object when multiple objects of the same category are present in the source image. A manual inspection of 100 randomly sampled triplets revealed only 4 minor counting inaccuracies (*e.g.*, number of sofa seats or drawers); and *(ii) Rotation data*: we encode the ground-truth camera rotation from Blender as a formatted string. We promote output diversity by selecting different system prompts and instruction wording during prompt construction.

[5] Blender is available under the GNU GPL License at https://www.blender.org/.

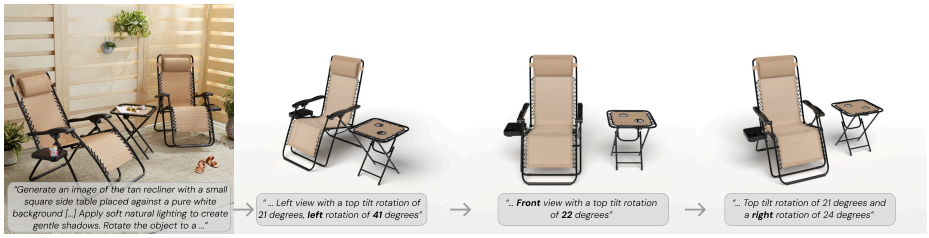


Fig. 3: A single 3D object rendered from three canonical viewpoints: *left*, *front*, and *right*. Each view includes the corresponding text prompt specifying the target orientation, along with precise elevation and azimuth annotations.

3.5 Human Quality Control

While the rendering process itself is deterministic, the resulting triplets may still contain quality issues that are difficult to detect automatically: *(i)* objects in the source image may be partially cropped or occluded, making attribute-preserving editing ill-defined; *(ii)* programmatic light placement in Blender can produce over- or under-exposed renders depending on material properties; and *(iii)* the product depicted in the source image may not correspond to the associated 3D model. We therefore perform manual quality control via a crowdsourcing platform to validate the final triplets. Further details are provided in the *Appendix* (Sec. 7.1.5). We summarize the dataset statistics in Table 1.

3.6 Statistics

ABO-Edit comprises 12,319 training and 400 validation samples spanning 20 product families. Chairs, sofas, and tables represent the most prevalent categories, with 2,800, 1,821, and 1,168 samples, respectively (see *Appendix* Fig. 7).

Camera Pose Distribution.
For each source image, we render three target views (front, left, and right) by positioning the camera at randomly sampled poses relative to the object's front-facing coordinate system. Specifically, the camera *elevation* (vertical angle above the horizontal plane) is sampled around 22°, providing a slight top-down perspective. The camera *azimuth* (horizontal rotation around the object) varies by view: front views use 0°, left views average 32.2°, while right views average 15.4° in the opposite direction (shown as absolute value in Table 1). We provide a visual example in Fig. 3. This asymmetry in azimuth ranges encourages diverse viewpoint coverage

Table 1: ABO-Edit statistics. We report the number of samples and the distribution of camera pose parameters (mean ± std) used during image rendering. Elevation is sampled uniformly within a fixed range, while azimuth varies by view direction.

Statistic	Train	Validation
Number of samples	12,319	400
Elevation angle	22.0° ± 0.8	22.0° ± 0.8
Azimuth (left view)	32.3° ± 7.2	32.5° ± 7.5
Azimuth (right view)	15.4° ± 5.8	15.2° ± 5.5

while maintaining natural viewing angles. Further details on product and rotation angle distributions are provided in the *Appendix* (Sec. 7.1.2 and Fig. 8).

4 Method

4.1 Preliminaries

Flow Models aim to transform a simple initial distribution p_1 into a complex data distribution p_0[6]. Formally, let $\mathbf{x}_0 \sim p_0$ denote a *data sample*, and $\mathbf{x}_1 \sim p_1$ denote a sample from the initial distribution, *i.e.*, the standard multivariate normal distribution $\mathcal{N}(0, \mathbf{I})$. A flow model defines a time-dependent probability path p_t for $t \in [0, 1]$ that interpolates between p_1 and p_0, governed by an ordinary differential equation (ODE):

$$\frac{\partial \mathbf{x}_t}{\partial t} = u_t^\theta(\mathbf{x}_t), \tag{1}$$

where $u_t^\theta(\mathbf{x}_t)$ is a time-dependent velocity field parameterized by a neural network. Sampling is performed by integrating this ODE from $t = 1$ to $t = 0$.

Rectified Flow [29,31] considers a linear probability path induced by interpolating between samples $\mathbf{x}_1$ and $\mathbf{x}_0$, yielding a constant velocity field:

$$\mathbf{x}_t = (1 - t)\,\mathbf{x}_0 + t\,\mathbf{x}_1, \tag{2}$$

$$\mathbf{v} = \frac{\partial \mathbf{x}_t}{\partial t} = \mathbf{x}_1 - \mathbf{x}_0. \tag{3}$$

The neural network predicts velocity $\mathbf{v}_\theta(\mathbf{x}_t, t)$, approximating the velocity field by minimizing the Conditional Flow Matching objective (CFM):

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_0,\, \mathbf{x}_1 \sim p_1,\, t \sim \mathcal{U}[0,1]} \left[\left\| \mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0) \right\|^2 \right]. \tag{4}$$

MultiModal Diffusion Transformer (MM-DiT) [14] extends the DiT architecture [35] by jointly modeling text and images in a shared latent space. Let $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ denote the target RGB image. A pretrained VAE encoder $\mathcal{E}$ maps the image $\mathbf{x}$ to a latent representation $\mathbf{z} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{h \times w \times c}$, where $h = H/r$, $w = W/r$, r denotes the spatial compression rate (typically 8), and c is the latent channel dimension. The latent $\mathbf{z}$ is then packed by merging 2×2 spatial patches into the channel dimension, yielding image tokens $\mathbf{z}_{\text{img}} \in \mathbb{R}^{N_{\text{img}} \times d}$, where N_{img} is the number of image tokens and d is the hidden dimension. Similarly, a text prompt is encoded using pretrained text encoders (*e.g.*, [39,40]) to obtain text tokens $\mathbf{z}_{\text{txt}} \in \mathbb{R}^{N_{\text{txt}} \times d}$, where N_{txt} is the sequence length. The noisy latent $\mathbf{z}_{\text{noisy}} \in \mathbb{R}^{N_{\text{img}} \times d}$ is constructed from $\mathbf{z}_{\text{img}}$ following Eq. 2.

For image-editing tasks, MM-DiT additionally conditions on an input RGB image $\mathbf{x}_{\text{cond}} \in \mathbb{R}^{H \times W \times 3}$. This conditioning image is processed through the same VAE encoder and packing operation to obtain latent $\mathbf{z}_{\text{cond}} \in \mathbb{R}^{N_{\text{cond}} \times d}$. The noisy and conditioning tokens are then concatenated along the sequence dimension to

[6] We use the inverted time convention, where $t = 0$ corresponds to a data sample.

form the transformer input, and then processed through L transformer blocks:

$$\mathbf{h}_0 = [\mathbf{z}_{\text{noisy}}; \mathbf{z}_{\text{cond}}] \in \mathbb{R}^{(N_{\text{img}}+N_{\text{cond}})\times d}, \tag{5}$$

$$\mathbf{h}_\ell = \text{TransformerBlock}_\ell(\mathbf{h}_{\ell-1}, \mathbf{z}_{\text{text}}, t), \quad \ell = 1, \ldots, L, \tag{6}$$

where t is the diffusion timestep. The last hidden state $\mathbf{h}_L \in \mathbb{R}^{(N_{\text{img}}+N_{\text{cond}})\times d}$ is split along the sequence dimension to recover the noisy and conditioning components:

$$\mathbf{h}_L = [\mathbf{h}_L^{\text{noisy}}; \mathbf{h}_L^{\text{cond}}], \quad \text{where } \mathbf{h}_L^{\text{noisy}} \in \mathbb{R}^{N_{\text{img}}\times d},\ \mathbf{h}_L^{\text{cond}} \in \mathbb{R}^{N_{\text{cond}}\times d}. \tag{7}$$

Finally, the velocity prediction $\mathbf{v}$ is computed solely from the noisy tokens via a learned projection:

$$\mathbf{v} = \text{Proj}(\mathbf{h}_L^{\text{noisy}}). \tag{8}$$

4.2 Observations

The predicted velocity $\mathbf{v}$ is computed solely from $\mathbf{h}_L^{\text{noisy}}$, while the conditioning output $\mathbf{h}_L^{\text{cond}}$ is discarded at inference. Yet $\mathbf{h}_L^{\text{cond}}$ remains coupled to the generation process: it participates in attention computations across all transformer layers and receives gradients during training. This raises a question: *what information does* $\mathbf{h}_L^{cond}$ *encode, and how does it evolve during denoising?* We investigate this through two complementary analyses.

Observation 1: Internal Representation. Fig. 4 reveals striking differences in how state-of-the-art image editing models utilize the conditioning latent $\mathbf{h}_L^{\text{cond}}$ throughout the denoising process. We decode $\mathbf{h}_L^{\text{cond}}$ using the VAE decoder $\mathcal{D}$ at various timesteps $t \in [0, 1]$, and observe two distinct behavioral patterns.

Models in the FLUX family [5,26] maintain representations of the conditioning image $\mathbf{x}_{\text{cond}}$ throughout denoising. In contrast, Qwen-based models [33,52] encode a prediction of the *final generated image* in $\mathbf{h}_L^{\text{cond}}$, even at high noise levels ($t \approx 1$). This behavior is particularly pronounced in Qwen-Image-Edit-2509-Lightning [33], a distilled, 8-step variant of Qwen-Image-Edit-2509 [52].

Observation 2: Velocity Prediction. From Eq. 3, we observe that rectified flow models predict a time-dependent velocity field corresponding to the displacement between data and noise along the linear interpolation path. At any time step t, we can recover an *estimate* of the clean image $\hat{\mathbf{x}}_{0,t}$ from the current state $\mathbf{x}_t$ via

$$\hat{\mathbf{x}}_{0,t} = \mathbf{x}_t - t\mathbf{v}_\theta(\mathbf{x}_t, t). \tag{9}$$

We provide the derivation in Sec. 7.4.1 (*Appendix*).

Following Eq. 9, in Fig. 5 (green curve) we plot the cosine similarity between the predicted clean image $\hat{\mathbf{x}}_{0,t}$ at timesteps t and the final generated image $\hat{\mathbf{x}}_0$. Embeddings are computed using the vision backbone of `CLIP-ViT-L/14` [39]. Even at high noise levels ($t = 1$), the similarity remains relatively high, indicating that the predicted velocity at early timesteps already closely approximates the final velocity. Note that this similarity naturally converges to 1 as $t \to 0$, since $\hat{\mathbf{x}}_{0,t}$ approaches $\hat{\mathbf{x}}_0$ by construction. A similar finding is reported in [27].

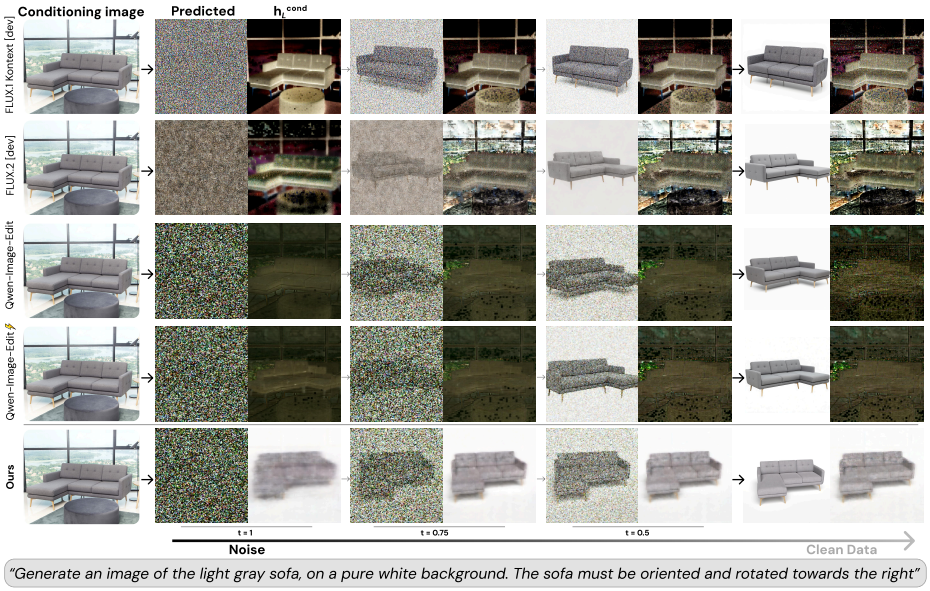


Fig. 4: Internal representations. We visualize both the predicted image and the decoded conditioning latent $\mathbf{h}_L^{\text{cond}}$ at different timesteps during denoising. Model evaluated *without* fine-tuning: *(i)* `FLUX.1 Kontext [dev]` [26]; *(ii)* `FLUX.2 [dev]` [5]; *(iii)* `Qwen-Image-Edit-2509-Lightning` [33] and *(iv)* `Qwen-Image-Edit-2509` [52]. We also visualize our method's internal representation after fine-tuning. FLUX models maintain representations of the conditioning image throughout denoising, whereas Qwen models *encode the final prediction* in $\mathbf{h}_L^{\text{cond}}$ even at high noise levels ($t \approx 1$). Note that our method encodes the prediction in the embedding space *while* preserving the original geometry.

4.3 FlowMirror: When Conditioning Mirrors the Generation

Motivation. *Observation* **1** reveals that certain model families naturally encode the final output in $\mathbf{h}_L^{\text{cond}}$, while *Observation* **2** shows that output predictions stabilize early in denoising. We hypothesize that explicitly supervising $\mathbf{h}_L^{\text{cond}}$ to match the target amplifies this beneficial behavior: it encourages the model to form accurate output representations early, while providing auxiliary gradient signal without additional parameters.

Based on these findings, we propose *FlowMirror*, a regularization objective that supervises the conditioning output $\mathbf{h}_L^{\text{cond}}$, typically not enforced during training. Crucially, our approach is *parameter-free* and incurs negligible computational overhead. Fig. 6 illustrates the overall framework.

Overview. Our key idea is to supervise the transformer's conditioning output $\mathbf{h}_L^{\text{cond}}$ using the VAE latent representation of the *target* image, $\mathbf{z}$. Operating in the VAE's compressed embedding space offers two advantages: *(i)* the lower-dimensional representation is computationally efficient, and *(ii)* all operations remain in latent space without invoking the decoder $\mathcal{D}$, avoiding substantial overhead.

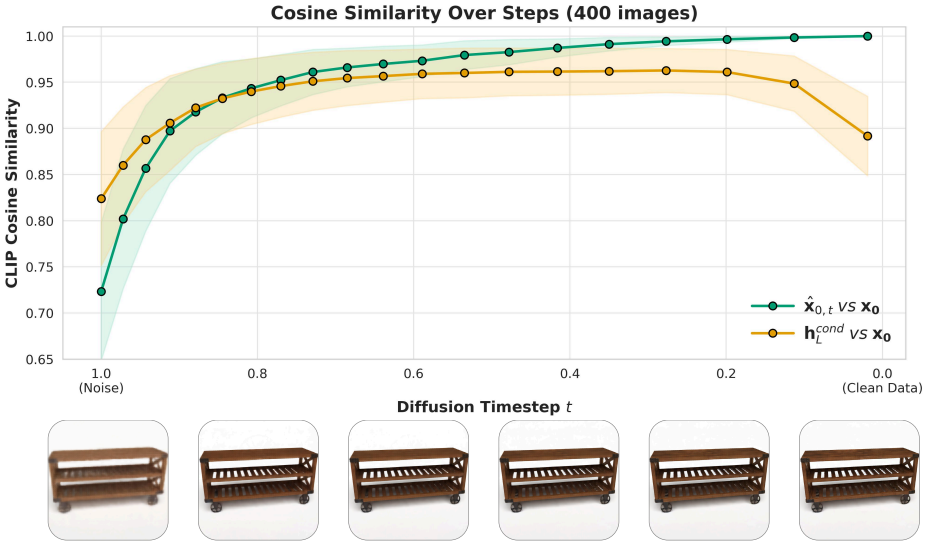


Fig. 5: Temporal evolution of predictions during denoising. Mean cosine similarity using CLIP embeddings on ABO-Edit validation set ($n = 400$). **Green**: similarity between *intermediate* prediction $\hat{\mathbf{x}}_{0,t}$ at timestep t and *final* output $\hat{\mathbf{x}}_0$. **Orange**: similarity between decoded conditioning embedding $\mathbf{h}_L^{\text{cond}}$ at timestep t and final output $\hat{\mathbf{x}}_0$ (boxes show a decoded sample at varying t). High similarity at early timesteps ($t \approx 1$) indicates that intermediate predictions already approximate the final generation.

To enable this supervision, we first reshape ("unpack") both representations into a shared spatial layout: $\mathbf{z} \in \mathbb{R}^{c\times h\times w}$ and $\mathbf{h}_L^{\text{cond}} \in \mathbb{R}^{c\times h\times w}$.

Multi-Scale Decomposition. We construct a Laplacian-pyramid–style decomposition over S scales to capture both fine details and global structure. Let $\mathbf{h_L}^{\text{cond}^{(s)}}$ and $\mathbf{z}^{(s)}$ denote representations at scale s, where each level applies a fixed, depthwise Gaussian blur $\mathcal{G}(\cdot)$ independently per channel.

High-Frequency (Detail Preservation). To encourage detail preservation, we penalize the ℓ_1 loss between the original and blurred tensors:

$$\mathcal{L}_{\text{HF}}^{(s)} = \left\| \left(\mathbf{h_L}^{\text{cond}^{(s)}} - \mathcal{G}(\mathbf{h_L}^{\text{cond}^{(s)}})\right) - \left(\mathbf{z}^{(s)} - \mathcal{G}(\mathbf{z}^{(s)})\right) \right\|_1 \tag{10}$$

Low-Frequency (Structure). Low-frequency components capture global structure. After Gaussian blurring, we downsample, via average pooling Down$(\cdot)$, and penalize discrepancy using an ℓ_2 loss:

$$\mathcal{L}_{\text{LF}}^{(s)} = \left\| \text{Down}\Big(\mathcal{G}(\mathbf{h_L}^{\text{cond}^{(s)}})\Big) - \text{Down}\Big(\mathcal{G}(\mathbf{z}^{(s)})\Big) \right\|_2^2 \tag{11}$$

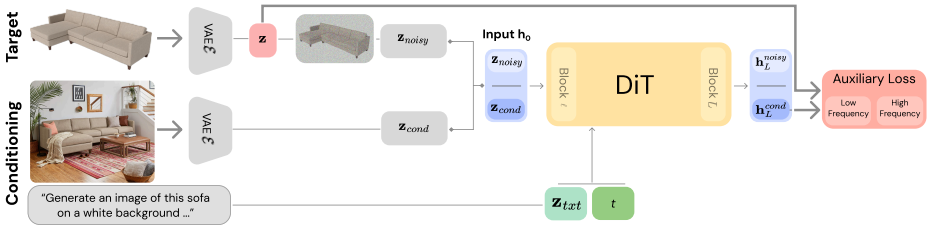


Fig. 6: Overview of the proposed auxiliary loss. Given noisy input $\mathbf{z}_{\text{noisy}}$ (*target* image), conditioning image embedding $\mathbf{z}_{\text{cond}}$, timestep t and text embeddings $\mathbf{z}_{\text{txt}}$, the transformer (DiT) predicts velocity v while also producing the conditioning output $\mathbf{h}_L^{\text{cond}}$. We supervise $\mathbf{h}_L^{\text{cond}}$ against the target latent $\mathbf{z}$ using a parameter-free, multi-scale decomposition that captures both fine details and global structure.

Total Loss. The full objective combines the standard CFM loss (Eq. 4) with our proposed auxiliary regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CFM}} + \lambda_{\text{aux}}\,\mathcal{L}_{\text{aux}}, \qquad \mathcal{L}_{\text{aux}} = \tfrac{1}{S}\sum_{s=1}^{S}\left(\mathcal{L}_{\text{LF}}^{(s)} + \lambda_{\text{HF}}\cdot\mathcal{L}_{\text{HF}}^{(s)}\right). \tag{12}$$

5 Experiment

Dataset. We evaluate on ABO-Edit, which provides challenging evaluation scenarios for visual object consistency, including: *(i)* precise viewpoint control with degree-level angular accuracy; *(ii)* 3D spatial transformations such as rotation and tilting; *(iii)* complex image editing requests combining product selection, background removal and pose request; and *(iv)* object-prompt alignment across visually similar products.

Implementation Details. We generate images at 1024×1024 resolution with each model's default configuration unless otherwise specified. Due to computational constraints and a resource-constrained setting, we train with a budget of 10k optimization steps on a single node equipped with 8 NVIDIA H100 GPUs, using an effective batch size of 32. We fine-tune using DoRA [30] with rank $r = 64$, learning rate $\eta = 10^{-4}$ and scaling factor $\alpha = 64$, maintaining a training resolution of 1024×1024. Training completes in approximately 43 hours (336 GPU-hours). For fine-tuned models, we evaluate three checkpoints trained with different seeds; for zero-shot methods, we use three different sampling seeds. We report mean and standard deviation in both cases. Empirically, we found the best performance with scales $S = 2$, $\lambda_{\text{HF}} = 0.2$, and $\lambda_{\text{aux}} = 0.01$. For dataset construction, we employ `Qwen3-VL-8B` [3] as VLM.

Baselines. Due to high computational requirements, we focus our fine-tuning comparison on `Qwen-Image-Edit-2509` [52] (in the following `Qwen-2509`) and `FLUX.1-Kontext-dev` [26]. We additionally report zero-shot performance of `FLUX.2-dev` [5], `Kandinsky 5.0` [1], and `Qwen-Image-Edit-2511` [52].

Table 2: Results on the ABO-Edit validation split. Indented rows denote addition of our auxiliary loss to each model. We train each fine-tuned model 3 times with different seeds to report mean ± std (sampling seeds for zero-shot). Best per section in **bold**.

Method	FID ↓	SSIM ↑	PSNR ↑	LPIPS ↓	CLIP ↑	DINOv2 ↑
Zero-shot baselines						
1) Kandinsky [1]	88.60 ±1.15	.644 ±.004	9.14 ±.08	.579 ±.007	83.55 ±.38	56.74 ±1.04
2) FLUX.1 [26]	59.32 ±.39	.789 ±.001	12.17 ±.06	.345 ±.003	90.76 ±.10	78.14 ±.25
3) FLUX.2 [5]	**53.67** ±.30	.803 ±.000	**12.80** ±.06	.320 ±.000	**92.66** ±.00	**83.67** ±.25
4) Qwen-2511 [52]	64.23 ±.47	.805 ±.001	12.73 ±.03	.336 ±.000	89.17 ±.22	77.83 ±.31
5) Qwen-2509 [52]	57.66 ±.02	**.818** ±.005	12.76 ±.14	**.312** ±.007	91.95 ±.07	82.30 ±.06
Fine-tuned on ABO-Edit (selected for best FID)						
6) FLUX.1 [26]	31.68 ±.18	.883 ±.002	18.62 ±.05	.149 ±.003	96.20 ±.10	90.99 ±.37
7) + *Aux Loss*	**31.26** ±.18	**.883** ±.001	**18.67** ±.05	**.148** ±.002	**96.30** ±.10	**91.25** ±.13
8) Qwen-2509 [52]	26.48 ±.86	.893 ±.001	19.64 ±.03	.112 ±.002	**97.21** ±.08	**93.98** ±.28
9) + *Aux Loss*	**25.91** ±.20	**.894** ±.001	**19.79** ±.08	**.111** ±.001	97.17 ±.05	93.94 ±.22

Evaluation Metrics. We evaluate generation quality using the Fréchet Inception Distance (`FID`) (↓) [19], structural fidelity and edge alignment with `SSIM` (↑) [51] and Peak-Signal-to-Noise Ratio (`PSNR` ↑), and perceptual similarity with `LPIPS` (↓) [59]. For semantic alignment, we compute `CLIP` (↑) [39] (`ViT-L/14@336px`) and `DINOv2` (↑) [34] (`ViT-B/14-reg`) image-image similarity scores. Metrics are computed using the TorchMetrics [12] package when available.

5.1 Can we leverage this space for supervision?

Table 2 summarizes all results on ABO-Edit. We first focus on zero-shot baselines (Rows 1–5), which highlight the inherent difficulty of the task. Qualitative inspection reveals recurring failure modes, including incomplete object segmentation, fog-like artifacts, and difficulty disambiguating target products from visually similar products. Rows 6–9 report results after fine-tuning on ABO-Edit, with checkpoints selected by validation FID over 3 separate training runs.

Predictive Latent Space. We first evaluate FlowMirror on models that *naturally* exhibit predictive behavior in the conditioning embedding space (Observation 1). As shown in Table 2 (rows 8–9), fine-tuning with the proposed parameter-free auxiliary loss improves `FID` (26.48 → **25.91**), alongside gains in `SSIM` (.893 → **.894**), `LPIPS` (.112 → **.111**), and `PSNR` (19.64 → **19.79**). These gains in pixel-level and perceptual similarity metrics indicate that our auxiliary loss encourages the model to better preserve fine-grained details and structural consistency. We note a negligible decrease in `CLIP` and `DINOv2`, indicating comparable semantic preservation. Models trained with the auxiliary loss also exhibit lower variance across most metrics, suggesting more stable optimization dynamics.

Non-Predictive Latent Space. We then evaluate FlowMirror on models that *do not* exhibit predictive behavior in the conditioning space (Fig. 4). As shown

in Table 2 (rows 6–7), the addition of the auxiliary loss consistently improves all metrics: FID (31.68 → **31.26**), SSIM (.883 → **.883**), LPIPS (.149 → **.148**), PSNR (18.62 → **18.67**), CLIP (96.20 → **96.30**), and DINOv2 (90.99 → **91.25**). This demonstrates that supervising the conditioning space is beneficial even when the model does not naturally encode target predictions, suggesting FlowMirror acts as an effective regularizer across rectified flow architectures.

Human Evaluation. Our method was preferred 4.4 percentage points more often than the baseline (see *Appendix* 7.3 for details).

5.2 Do Generative Models Have Spatial Awareness?

We investigate whether editing models can accurately orient objects to specified viewpoints given precise rotation angles expressed in natural language. We partition the validation set into three orientation buckets based on ground-truth target azimuth available in ABO-Edit: *left*, *front*, and *right*. For each generated image (using ABO-Edit validation prompts), we estimate the relative orientation with respect to the target using `Orient-Anything-v2` [49]. To ensure measurement reliability, we retain only samples with an "exact" predicted symmetry type (using `Orient-Anything-v2`), and discard those with azimuth errors exceeding 90°, which indicate orientation estimation failures.

Table 3: Azimuth error (°) between generated and target images on the validation set, bucketed by viewpoint. Lower is better. Results shown as mean ± std.

Model	Left	Front	Right
Zero-shot baselines			
1) Kandinsky [1]	47.2 ±25.8	21.0 ±22.0	37.2 ±26.4
2) FLUX.1 [26]	21.7 ±22.0	23.7 ±17.7	38.9 ±20.6
3) FLUX.2 [5]	48.2 ±25.2	11.9 ±13.6	42.2 ±21.0
4) Qwen-2511 [52]	46.3 ±26.5	18.7 ±20.4	30.5 ±17.2
5) Qwen-2509 [52]	39.1 ±28.0	18.5 ±15.1	31.7 ±17.2
Fine-tuned on ABO-Edit			
6) **Qwen-2509**	**2.4** ±10.2	**1.0** ±2.2	**1.2** ±0.9

Results. Table 3 reports the azimuth error distribution across viewpoints. Fine-tuning on ABO-Edit yields substantial improvements in spatial control, achieving mean errors below 2.5°. This shows that task-specific training enables precise object orientation at degree-level accuracy without architectural changes. In contrast, zero-shot models exhibit considerably larger errors (rows 1–5), indicating that fine-grained angular placement from text prompts alone remains challenging without task-specific adaptation. We also observe that some models show asymmetric performance across left and right views, which may reflect biases in their pre-training data distributions.

Spatial Control. To demonstrate precise spatial control, we generate 5 novel views at varying rotation angles. As shown in Fig. 11 (*Appendix*), our fine-tuned model achieves fine-grained angular control while maintaining strong visual consistency across generated views.

Failure Modes. As shown in Fig. 12 (*Appendix*), zero-shot models exhibit bimodal error distributions across viewpoint buckets. This bimodality suggests two distinct failure modes: *(i)* the model correctly interprets the target orientation but introduces angular errors, and *(ii)* the model misinterprets the requested

viewpoint entirely, rendering an opposing orientation (*e.g.*, right-facing instead of left-facing). We provide per-model breakdowns with specific mode values in the supplementary material. To ground these errors in concrete examples, we illustrate representative failure cases in Fig. 13 (*Appendix*).

5.3 Can Conditioning Embeddings Outpace Velocity Predictions?

Fig. 5 (orange) reports the similarity between the conditioning embedding $\mathbf{h}_L^{\text{cond}}$ at timestep t and the final generated image $\hat{\mathbf{x}}_0$, computed by decoding $\mathbf{h}_L^{\text{cond}}$ with decoder $\mathcal{D}$ and measuring CLIP [39] similarity. Surprisingly, with our auxiliary loss, the model's predicted image in the conditioning space exhibits higher similarity to the target than the predicted velocity (green) at noisy timesteps, successfully encoding the target image. At high noise ($t = 1$), this gap is substantial (0.823 *vs*. 0.723). Shape, structure, and overall quality are already encoded at early timesteps, leaving little room for the model to alter these properties.

5.4 Analysis and Ablations

As shown in Table 2 (rows 6–9), adding $\mathcal{L}_{\text{aux}}$ improves generation quality. Due to computational constraints, we now ablate its design choices on `Qwen-Image-Edit-2509` using a single fixed seed. We initially explored matching VAE channel-wise mean and std of $\mathbf{h}_L^{\text{cond}}$ to $\mathbf{z}$, but this worsens `FID` (25.18 $\rightarrow$ 26.15). We attribute this to statistics conflicting with the spatially-grounded supervision of the loss. Normalizing $\mathcal{L}_{\text{aux}}$ by the number of scales further helps (26.52 $\rightarrow$ 26.15). Reducing λ_{aux} from 0.1 to 0.01 improves `FID` (25.93 $\rightarrow$ 25.55), indicating lighter regularization avoids interfering with the CFM objective. Removing the HF term entirely ($\lambda_{\text{HF}} = 0$) yields a high `FID` among variants (27.07), confirming that high-frequency supervision is essential for detail preservation. However, over-weighting it ($\lambda_{\text{HF}} = 1.0$) also degrades performance relative to $\lambda_{\text{HF}} = 0.2$, as excessive detail pressure interferes with the LF structural signal. Increasing beyond $S = 2$ degrades performance, likely because finer pyramid levels lose meaningful structure through excessive downsampling.

6 Conclusion & Impact

We presented three contributions. First, *ABO-Edit*, a benchmark of 12k+ triplets pairing real-world product photographs with studio-quality targets rendered from artist-designed 3D assets, enabling controlled evaluation of geometry, texture, and fine-grained details. Second, we revealed that rectified flow models encode target predictions in the conditioning embedding space $\mathbf{h}_L^{\text{cond}}$ even at high noise levels, and that these representations stabilize early in denoising. Third, we proposed *FlowMirror*, a parameter-free auxiliary loss that supervises this embedding space, improving generation quality across multiple metrics on two architectures.

Limitations. A domain gap exists between our Blender-rendered targets and real photographs, particularly in material appearance and lighting. Our fine-tuning experiments were limited to DoRA adaptation with a 10k-step budget; full fine-tuning may yield further gains. Finally, evaluation is restricted to ABO-Edit.
Future Work. An important direction for future work is to close the synthetic–real gap via photorealistic rendering or using real image pairs, to expand to diverse product types, and to investigate FlowMirror's generalization to additional models and tasks.
Impact. Our analysis opens new directions for future supervision strategies beyond CFM training, as demonstrated by FlowMirror's improvements. ABO-Edit further enables research on fine-grained object preservation and degree-level viewpoint control (Fig. 11, *Appendix*), as well as the inverse task of generating lifestyle imagery from studio photographs.

References

1. Arkhipkin, V., Korviakov, V., Gerasimenko, N., Parkhomenko, D., Vasilev, V., Letunovskiy, A., Vaulin, N., Kovaleva, M., Kirillov, I., Novitskiy, L., Koposov, D., Kiselev, N., Varlamov, A., Mikhailov, D., Polovnikov, V., Shutkin, A., Agafonova, J., Vasiliev, I., Kargapoltseva, A., Dmitrienko, A., Maltseva, A., Averchenkova, A., Kim, O., Nikulina, T., Dimitrov, D.: Kandinsky 5.0: A Family of Foundation Models for Image and Video Generation (2025), https://arxiv.org/abs/2511.14993 3, 11, 12, 13, 27
2. Bai, J., Chow, W., Yang, L., Li, X., Li, J., Zhang, H., Yan, S.: HumanEdit: A High-Quality Human-Rewarded Dataset for Instruction-based Image Editing (2025), https://arxiv.org/abs/2412.04280 25
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-VL Technical Report (2025), https://arxiv.org/abs/2511.21631 11
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-VL Technical Report (2025), https://arxiv.org/abs/2502.13923 4
5. Black Forest Labs: FLUX.2. https://bfl.ai/blog/flux-2 (Jun 2025), accessed: 2026 2, 3, 4, 8, 9, 11, 12, 13, 27
6. Blender Online Community: Blender - a 3D Modelling and Rendering Package (2025), www.blender.org 3
7. Brooks, T., Holynski, A., Efros, A.A.: InstructPix2Pix: Learning To Follow Image Editing Instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18392–18402 (June 2023), https://openaccess.thecvf.com/content/CVPR2023/html/Brooks_InstructPix2Pix_Learning_To_Follow_Image_Editing_Instructions_CVPR_2023_paper.html 25

8. Brylla, D., Walsh, G.: Scene Sells: Why Spatial Backgrounds Outperform Isolated Product Depictions Online. International Journal of Electronic Commerce **24**(4), 497–526 (Oct 2020). https://doi.org/10.1080/10864415.2020.1806470, http://dx.doi.org/10.1080/10864415.2020.1806470 4
9. Cai, Q., Chen, J., Chen, Y., Li, Y., Long, F., Pan, Y., Qiu, Z., Zhang, Y., Gao, F., Xu, P., Wang, Y., Yu, K., Chen, W., Feng, Z., Gong, Z., Pan, J., Peng, Y., Tian, R., Wang, S., Zhao, B., Yao, T., Mei, T.: HiDream-I1: A High-Efficient Image Generative Foundation Model with Sparse Diffusion Transformer (2025), https://arxiv.org/abs/2505.22705 4
10. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22560–22570 (October 2023), https://openaccess.thecvf.com/content/ICCV2023/html/Cao_MasaCtrl_Tuning-Free_Mutual_Self-Attention_Control_for_Consistent_Image_Synthesis_and_ICCV_2023_paper.html 2
11. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., Guillaumin, M., Malik, J.: ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 21094–21104. IEEE (Jun 2022). https://doi.org/10.1109/cvpr52688.2022.02045, http://dx.doi.org/10.1109/cvpr52688.2022.02045 2, 4, 5
12. Detlefsen, N.S., Borovec, J., Schock, J., Jha, A.H., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M., Falcon, W.: TorchMetrics - Measuring Reproducibility in PyTorch. Journal of Open Source Software **7**(70), 4101 (2022), https://doi.org/10.21105/joss.04101 12
13. Di, W., Sundaresan, N., Piramuthu, R., Bhardwaj, A.: Is a Picture Really Worth a Thousand Words? - On the Role of images in E-commerce. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining. p. 633–642. WSDM '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2556195.2556226, https://doi.org/10.1145/2556195.2556226 4
14. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, JMLR.org (2024), https://dl.acm.org/doi/10.5555/3692070.3692573 4, 7
15. Fan, X., Zhang, C., Yang, Y., Shang, Y., Zhang, X., He, Z., Xiao, Y., Long, B., Wu, L.: Automatic Generation of Product-Image Sequence in E-commerce. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 2851–2859. KDD '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3534678.3539149, https://doi.org/10.1145/3534678.3539149 2, 4
16. Google DeepMind: Gemini 2.5 Flash Image Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf (Sep 2025), model card published/updated September 2025 2, 4
17. Google DeepMind: Gemini 3 Pro Image Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Image-Model-Card.pdf (Nov 2025), model card published November 2025 2, 4
18. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-Prompt Image Editing with Cross-Attention Control. In: The Eleventh

International Conference on Learning Representations (2023), https://openreview.net/forum?id=_CDixzkzeyb 2, 25
19. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf 12
20. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen Video: High Definition Video Generation with Diffusion Models (2022), https://arxiv.org/abs/2210.02303 4
21. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf 3, 4
22. Ho, J., Salimans, T.: Classifier-Free Diffusion Guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021), https://openreview.net/forum?id=qw8AKxfYbI 3
23. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video Diffusion Models. In: Advances in Neural Information Processing Systems. vol. 35, pp. 8633–8646. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/39235c56aef13fb05a6adc95eb9d8d66-Paper-Conference.pdf 4
24. Hui, M., Yang, S., Zhao, B., Shi, Y., Wang, H., Wang, P., Xie, C., Zhou, Y.: HQ-Edit: A High-Quality Dataset for Instruction-based Image Editing. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=mZptYYttFj 25
25. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the Design Space of Diffusion-Based Generative Models. In: Advances in Neural Information Processing Systems. vol. 35, pp. 26565–26577. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/a98846e9d9cc01cfb87eb694d946ce6b-Paper-Conference.pdf 3
26. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space (2025), https://arxiv.org/abs/2506.15742 2, 3, 4, 8, 9, 11, 12, 13, 27
27. Li, S., Gao, M., Liu, Y., Lin, Z., Wang, F., Dai, F.: TReFT: Taming Rectified Flow Models For One-Step Image Translation. arXiv preprint arXiv:2511.20307 (2025) 8
28. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3D: High-Resolution Text-to-3D Content Creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 300–309 (June 2023), https://openaccess.thecvf.com/content/CVPR2023/html/Lin_Magic3D_High-Resolution_Text-to-3D_Content_Creation_CVPR_2023_paper.html 2
29. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow Matching for Generative Modeling. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=PqvMRDCJT9t 4, 7
30. yang Liu, S., Wang, C.Y., Yin, H., Molchanov, P., Wang, Y.C.F., Cheng, K.T., Chen, M.H.: DoRA: Weight-Decomposed Low-Rank Adaptation. In: Forty-first

International Conference on Machine Learning (2024), https://openreview.net/forum?id=3d5CIRG1n2 11
31. Liu, X., Gong, C., qiang liu: Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=XVjTT1nw5z 4, 7
32. Maier, E.: The Negative Effect of Product Image Inconsistency on Product Overviews During the Online Product Search. International Journal of Electronic Commerce **23**(1), 110–143 (Jan 2019). https://doi.org/10.1080/10864415.2018.1512281, http://dx.doi.org/10.1080/10864415.2018.1512281 4
33. ModelTC: Qwen-image-lightning. https://github.com/ModelTC/Qwen-Image-Lightning (2024), accessed: 2026 8, 9
34. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision. Transactions on Machine Learning Research (2024), https://openreview.net/forum?id=a68SUt6zFt, featured Certification 12
35. Peebles, W., Xie, S.: Scalable Diffusion Models with Transformers. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4172–4182 (2023). https://doi.org/10.1109/ICCV51070.2023.00387 4, 7
36. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=di52zR8xgf 2, 4
37. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFusion: Text-to-3D using 2D Diffusion. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=FjNys5c7VyY 4
38. Qian, Y., Bocek-Rivele, E., Song, L., Tong, J., Yang, Y., Lu, J., Hu, W., Gan, Z.: Pico-Banana-400K: A Large-Scale Dataset for Text-Guided Image Editing (2025), https://arxiv.org/abs/2510.19808 25
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/radford21a.html 4, 5, 7, 8, 12, 14
40. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html 4, 7
41. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical Text-Conditional Image Generation with CLIP Latents (2022), https://arxiv.org/abs/2204.06125 4
42. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis With Latent Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022), https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper 3, 4

43. Sheynin, S., Polyak, A., Singer, U., Kirstain, Y., Zohar, A., Ashual, O., Parikh, D., Taigman, Y.: Emu Edit: Precise Image Editing via Recognition and Generation Tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8871–8879 (June 2024), https://openaccess.thecvf.com/content/CVPR2024/html/Sheynin_Emu_Edit_Precise_Image_Editing_via_Recognition_and_Generation_Tasks_CVPR_2024_paper.html 25
44. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 2256–2265. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/sohl-dickstein15.html 3
45. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=St1giarCHLP 3
46. Team, I., Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., Li, Z., Li, Z.Y., Liu, D., Liu, D., Shi, J., Wu, Q., Yu, F., Zhang, C., Zhang, S., Zhou, S.: Z-Image: An Efficient Image Generation Foundation Model with Single-Stream Diffusion Transformer (2025), https://arxiv.org/abs/2511.22699 4
47. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and Advanced Large-Scale Video Generative Models (2025), https://arxiv.org/abs/2503.20314 4
48. Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D.J., Soricut, R., Baldridge, J., Norouzi, M., Anderson, P., Chan, W.: Imagen Editor and EditBench: Advancing and Evaluating Text-Guided Image Inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18359–18369 (June 2023), https://openaccess.thecvf.com/content/CVPR2023/html/Wang_Imagen_Editor_and_EditBench_Advancing_and_Evaluating_Text-Guided_Image_Inpainting_CVPR_2023_paper.html 25
49. Wang, Z., Zhang, Z., Xu, J., Wang, J., Pang, T., Du, C., Zhao, H., Zhao, Z.: Orient Anything V2: Unifying Orientation and Rotation Understanding. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=n3armuTFit 13
50. Wang, Z., Bao, J., Gu, S., Chen, D., Zhou, W., Li, H.: DesignDiffusion: High-Quality Text-to-Design Image Generation with Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20906–20915 (June 2025), https://openaccess.thecvf.com/content/CVPR2025/html/Wang_DesignDiffusion_High-Quality_Text-to-Design_Image_Generation_with_Diffusion_Models_CVPR_2025_paper.html 2
51. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image Quality Assessment: from Error Visibility to Structural Similarity. IEEE Transactions on Image Processing

13(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861, https://doi.org/10.1109/TIP.2003.819861 12

52. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-Image Technical Report (2025), https://arxiv.org/abs/2508.02324 2, 3, 4, 8, 9, 11, 12, 13, 26, 27, 28
53. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 Technical Report (2025), https://arxiv.org/abs/2505.09388 23
54. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.H.: Diffusion Models: A Comprehensive Survey of Methods and Applications. ACM Comput. Surv. **56**(4) (Nov 2023). https://doi.org/10.1145/3626235, https://doi.org/10.1145/3626235 3
55. Ye, J., Wang, P., Li, K., Shi, Y., Wang, H.: Consistent-1-to-3: Consistent Image to 3D View Synthesis via Geometry-aware Diffusion Models . In: 2024 International Conference on 3D Vision (3DV). pp. 664–674. IEEE Computer Society, Los Alamitos, CA, USA (Mar 2024). https://doi.org/10.1109/3DV62453.2024.00027, https://doi.ieeecomputersociety.org/10.1109/3DV62453.2024.00027 2
56. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: ImgEdit: A Unified Image Editing Dataset and Benchmark. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2025), https://openreview.net/forum?id=uUCSrMlfD3 25
57. Yu, J.J., Forghani, F., Derpanis, K.G., Brubaker, M.A.: Long-Term Photometric Consistent Novel View Synthesis with Diffusion Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7094–7104 (October 2023), https://openaccess.thecvf.com/content/ICCV2023/html/Yu_Long-Term_Photometric_Consistent_Novel_View_Synthesis_with_Diffusion_Models_ICCV_2023_paper.html 2
58. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In: Advances in Neural Information Processing Systems. vol. 36, pp. 31428–31449. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/64008fa30cba9b4d1ab1bd3bd3d57d61-Paper-Datasets_and_Benchmarks.pdf 25
59. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018), https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html 12
60. Zhang, Z., Xie, J., Lu, Y., Yang, Z., Yang, Y.: Enabling Instructional Image Editing with In-Context Generation in Large Scale Diffusion Transformer. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=5WyqKH9nOS 4

61. Zhao, H., Ma, X., Chen, L., Si, S., Wu, R., An, K., Yu, P., Zhang, M., Li, Q., Chang, B.: UltraEdit: Instruction-based Fine-Grained Image Editing at Scale. In: Advances in Neural Information Processing Systems. vol. 37, pp. 3058–3093. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-0100, https://proceedings.neurips.cc/paper_files/paper/2024/file/05a30a0fc9e6bacdd3abd4ca8508a9e6-Paper-Datasets_and_Benchmarks_Track.pdf 25

7 Appendix.

In this appendix, we provide additional details organized as follows:

- **ABO-Edit dataset** (Sec. 7.1): we present the static prompt used for dataset construction (Sec. 7.1.1), analyze product type distribution (Sec. 7.1.2), describe the angle distribution during target image rendering (Sec. 7.1.3), and report image resolution statistics (Sec. 7.1.4). Finally, we provide details on human quality control (Sec. 7.1.5) and provide a discussion of related datasets (Sec. 7.1.6).
- **Spatial Awareness** (Sec. 7.2): we first show the fine-grained control and spatial understanding of our method (Sec. 7.2.1). Then, we analyze the distribution of azimuth error between generated and target images across fine-tuned and zero-shot models (Sec. 7.2.2), and visualize examples of such failures (Sec. 7.2.3).
- **Human Evaluation** (Sec. 7.3): we report details regarding the human evaluation for the blind pairwise comparison between our method and the baseline.
- **Flow Matching** (Sec. 7.4): we derive the $\hat{\mathbf{x}}_0$ estimation used in our formulation.

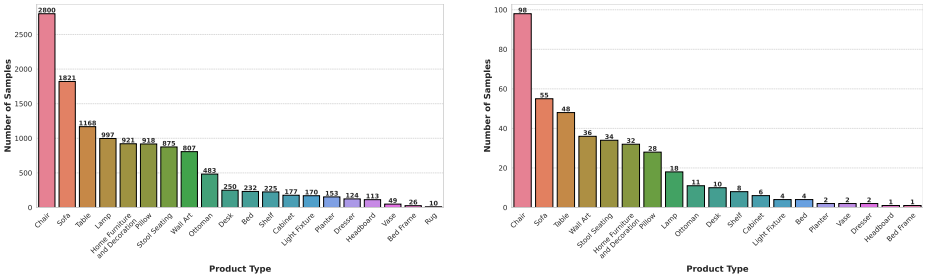
7.1 ABO-Edit

7.1.1 Prompt for Dataset Construction. Template Prompt (dataset generation):

```
Generate an image of the {OBJECT_DESC} on a pure white background.
Ensure strict preservation of the {OBJECT_CATEGORY}'s precise
geometry, edges, intersections, and proportions.
Remove people, info-graphics, and text.
The entire {OBJECT_CATEGORY} is fully visible (not cropped) and
isolated, no other objects or text in the scene.
Ensure a soft natural lighting with gentle shadows.
{ROTATION_INFORMATION}. The rotation coordinate system is the
front-view of the object.
```

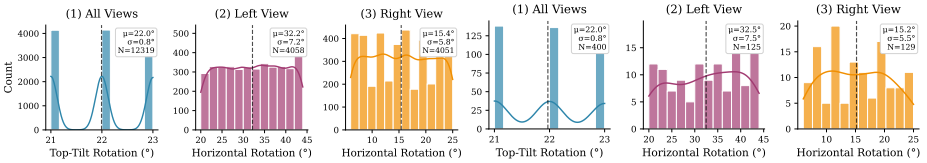
7.1.2 Product Types Distribution. Fig. 7 presents the distribution of product types across dataset splits. The training set (Fig. 7a) contains 12,319 images, while the validation set (Fig. 7b) comprises 400 samples. Both splits exhibit a long-tailed distribution, with certain product categories being significantly more prevalent than others.

7.1.3 Rotation Examples. In Fig. 8, we show the rotation distribution of the train and validation set. We report the distribution of rotation angles across dataset splits (mean ± std), bucketed by object rotation.



(a) Training set. **(b)** Validation set.

Fig. 7: Distribution of product types across dataset splits.



(a) Rotation angle distribution (*training* set). **(b)** Rotation angle distribution (*validation* set).

Fig. 8: Distribution of rotation angles across dataset splits (mean ± std), bucketed by object rotation. (**a**-1, **b**-1) show the tilt angle distribution across all views. (**a**-2/3, **b**-2/3) show the azimuth angle distribution for the *left* and *right* views specifically. The *front* view is omitted as its azimuth is 0° by definition.

We first report the tilt angle across all views (*left*, *front*, and *right*), with a mean tilt rotation of 22° (**a**-1, **b**-1). Then, we report the azimuth angle for the *left* and *right* views (**a**-2/3, **b**-2/3); the *front* view has a fixed azimuth of 0°. For the *left* view, we sample angles uniformly between 20° and 45°, mimicking conventional e-commerce product placement. For the *right* view, we sample angles between 5° and 25°, representing a narrower range. Furthermore, Fig. 9 shows representative samples from our proposed ABO-Edit. The source images depict challenging scenarios with products in use, while the corresponding renders illustrate high-quality outputs across three distinct viewpoints per object.

7.1.4 Image Resolution Statistics. ABO-Edit comprises high-resolution images suitable for fine-grained visual analysis, as shown in Fig. 10. The training set contains 12,319 source images with a mean resolution of 2361 × 2338 pixels (width × height), while the validation set contains 400 source images with comparable statistics (mean: 2373 × 2353 pixels). The high average resolution enables detailed object-level analysis and supports downstream tasks requiring fine-grained visual features.

7.1.5 Human Quality Control. First, to minimize annotator cognitive load, we use an LLM (`Qwen3-4B` [53]) to shorten the prompts $p_{\text{src}\to\text{trg}}$ by removing

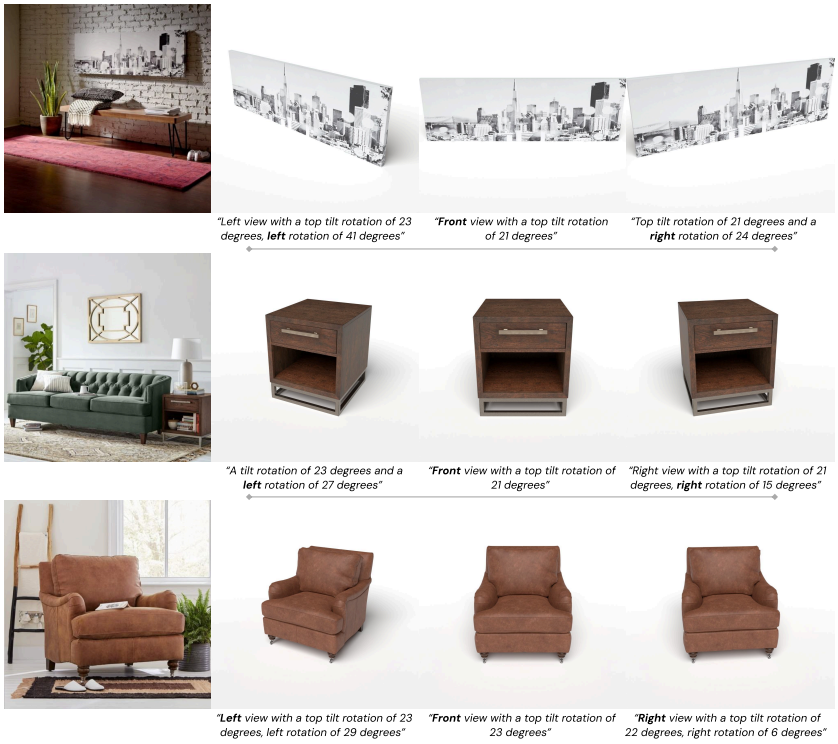


Fig. 9: We show representative samples for each of the three views: *front*, *left*, and *right*. For each sample, we also display the rotation parameters, specifying the target view and the angle by which the object is rotated.

unnecessary details while retaining the core information, *i.e.*, object description, background specification, and rotation parameters. We then define the following rejection criteria: *(i) Object Visibility*: the object in the source image is partially cropped or occluded; *(ii) Geometry Artifacts*: the object's shape, structure, or proportions are not preserved in the target image; *(iii) Prompt Mismatch*: the rotation of the product in the target image does not match the rotation specified in the prompt; *(iv) Appearance Artifacts*: material details, fabric patterns, or colors are inconsistent between source and target, or the rendered object exhibits unnatural shadows or incorrect exposure; and *(v) Hallucinated Object*: the rendered product does not correspond to the product depicted in the source image. We proceed by designing an HTML annotation template containing the task guidelines along with representative examples of accepted and rejected triplets. The template was iteratively refined over several pilot rounds to improve annotator consensus. After finalizing the guidelines, we perform full-scale annotation.

To evaluate annotation quality, we manually label a reference set of 500 triplets, which we treat as ground truth. We compare several label aggregation

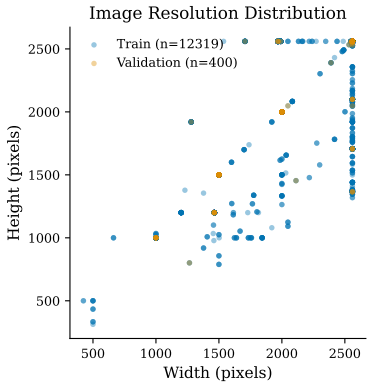


Fig. 10: Source images resolution distribution across train (blue) and validation (orange) splits.

strategies against this set and find that majority vote yields the highest agreement, achieving high precision and recall on this held-out validation set.

7.1.6 Related Works: Datasets. Several benchmarks have been proposed for text-guided image editing, varying in scale, annotation methodology, and task coverage. EditBench [48] introduced a text-guided image inpainting benchmark comprising 240 images with masked-region annotations. InstructPix2Pix [7] pioneered large-scale synthetic editing pair generation by combining GPT-3 and Prompt-to-Prompt [18]. HQ-Edit [24] proposed a data collection pipeline that leverages advanced foundation models to improve editing quality. MagicBrush [58] provides 10K human-annotated triplets of source images, editing instructions, and target images. Similarly, Emu-Edit [43] proposes a benchmark that includes seven different image editing tasks, including object removal, object addition, and localized alteration. UltraEdit [61] proposes a massive 4M-sample dataset for instruction-based image editing, enabling support for region-based instruction. More recently, Pico-Banana-400k [38] introduced a 400K synthetic dataset covering both single- and multi-turn editing, while ImgEdit [56] proposed a large-scale counterpart spanning diverse edit types such as background change, motion change, and style transfer. HumanEdit [2] contributes 5,751 manually constructed editing pairs, emphasizing annotation quality through dedicated human annotators.

Positioning of our work. Contrary to the works discussed above, we present a benchmark specifically designed to evaluate visual object consistency. While existing datasets focus on general-purpose editing tasks (*e.g.*, object removal, style transfer, or inpainting), ABO-Edit provides pixel-accurate target images rendered from artist-designed 3D assets, guaranteeing that geometry, texture, and material are preserved by construction. ABO-Edit requires models to perform a sequence of challenging sub-tasks: disambiguating and selecting a target product from natural language among visually similar candidates within a single image,

removing backgrounds, generating physically plausible shadows, and executing 3D rotations with degree-level angular precision. Furthermore, ABO-Edit additionally offers degree-level orientation annotations (with multi-view samples), enabling systematic evaluation of spatial consistency under 3D transformations, a capability absent from all prior benchmarks.

7.2 Do Generative Models Have Spatial Awareness?

In this section, we provide additional analysis on the spatial awareness of generative models.

7.2.1 Fine-grained Rotation Control. Fig. 11 illustrates the fine-grained rotation control achieved by our fine-tuned model. The top row shows a brown cabinet as the source image; we then generate novel views at varying angles using our model (based on `Qwen-Image-Edit-2509` [52]) while keeping all other attributes fixed. The bottom row repeats the experiment with a different source image, *i.e.* a brown sofa. The results demonstrate precise viewpoint control and strong appearance consistency across generated views, with challenging structure preservation.

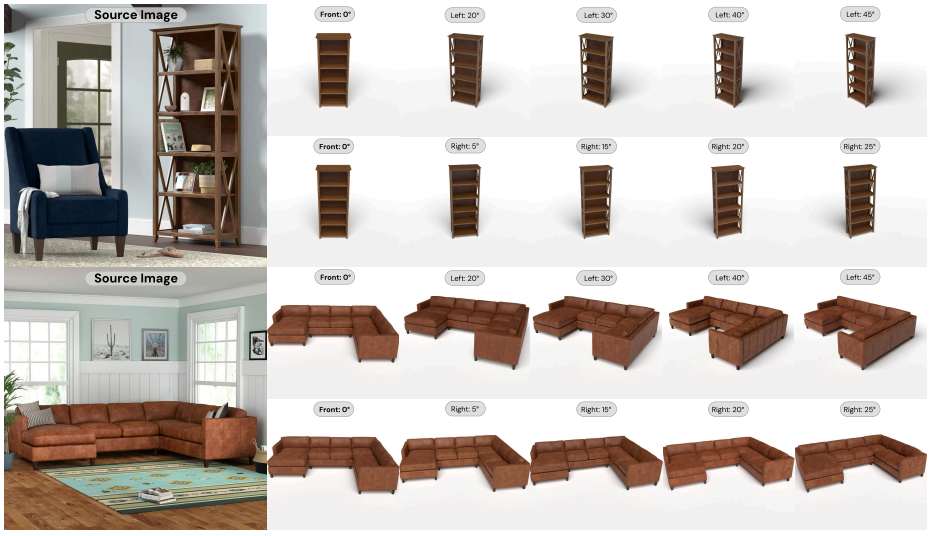
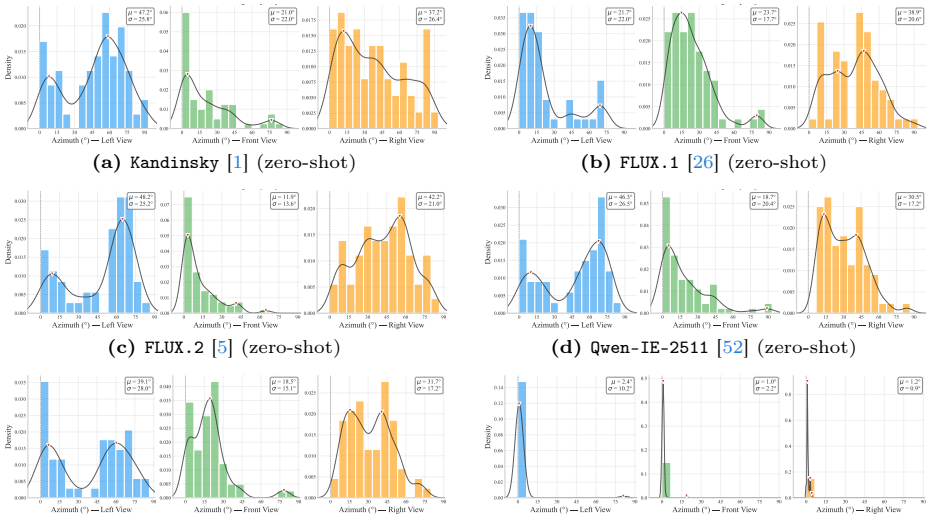


Fig. 11: Fine-grained rotation control. Given a source image (leftmost column), our model generates novel views at specified rotation angles while preserving object identity and appearance. Results are shown for two distinct objects: a shelf (top) and a sofa (bottom). For visualization clarity, we display outputs at 5° intervals; however, our model supports continuous viewpoint manipulation at 1° granularity, enabling precise and consistent control across diverse object categories.



(a) `Kandinsky` [1] (zero-shot) **(b)** `FLUX.1` [26] (zero-shot)

(c) `FLUX.2` [5] (zero-shot) **(d)** `Qwen-IE-2511` [52] (zero-shot)

(e) `Qwen-IE-2509` [52] (zero-shot) **(f)** `Qwen-IE-2509` [52] **(fine-tuned)**

Fig. 12: Azimuth error distributions across viewpoints (*left*, *front*, *right*) for various models. (a–e) Zero-shot baselines exhibit broad, often bimodal distributions indicating frequent orientation errors. (f) Fine-tuning on ABO-Edit yields sharply peaked distributions near zero, demonstrating precise viewpoint control.

7.2.2 Azimuth Distribution. In Fig. 12, we report the distribution of azimuth error between the generated and target images, across fine-tuned and zero-shot models. From the figure, we can derive the following observations:

(i) Zero-shot models show higher errors on this task: the *left* view yields a mean error of 40.5°, *front* 18.76°, and *right* 36.1°, indicating that fine-grained spatial placement from text prompts alone remains challenging without task-specific adaptation.
(ii) For one model [26], the *left* view azimuth error is nearly half that of the *right* view, suggesting a potential bias in its training data distribution.
(iii) Nearly all models exhibit bimodal error distributions, characterized by high mean azimuth errors in object placement. Furthermore, these models frequently misinterpret the target viewpoint altogether, further underscoring the difficulty of this task without task-specific training.

We ground such errors in visual examples in the following Section.

7.2.3 Rotation Failures. Fig. 13 illustrates representative orientation failures. When evaluated zero-shot on our validation set, one representative model [52] shows limited spatial controllability on this task: left rotation prompts frequently yield right-facing views, and vice versa, suggesting the model lacks a coherent internal representation of 3D orientation.

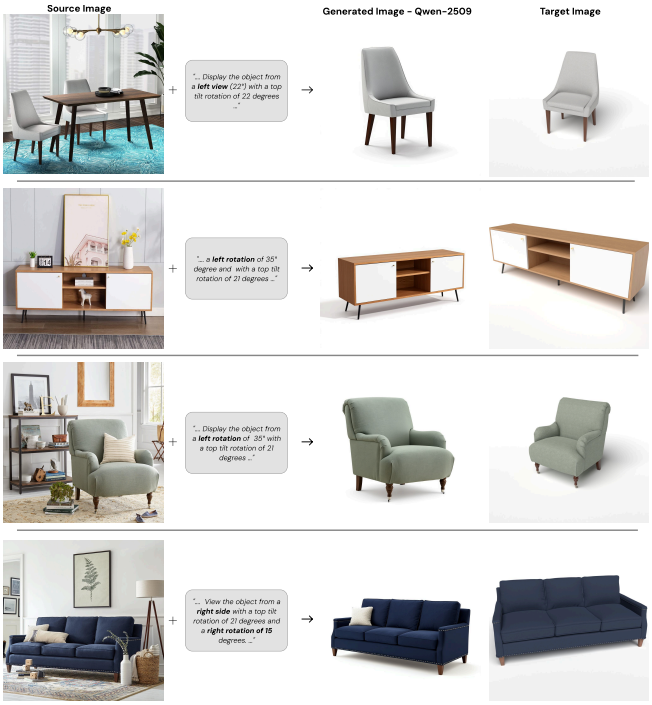


Fig. 13: Representative failure cases for spatial reasoning on our validation set. Each triplet shows: source lifestyle image, zero-shot model output [52], and expected target with correct orientation.

7.3 Human Evaluation

We conducted a blind pairwise comparison between our method and the baseline on 278 images. Human annotators were asked to select the preferred result, and the final preference was determined by majority vote. Our method was preferred 4.4% more often than the baseline.

7.4 Flow Matching

7.4.1 $\hat{\mathbf{x}}_0$ Derivation. During inference, at any time step t, we can retrieve the clean image $\hat{\mathbf{x}}_0$ from the current state $\mathbf{x}_t$ via: $\hat{\mathbf{x}}_0 = \mathbf{x}_t - t\mathbf{v}_\theta(\mathbf{x}_t, t)$.
Derivation. During training, the ground truth velocity is defined as:

$$\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0.$$

We start by reporting and then rewriting Eq. 2:

$$\begin{aligned}\mathbf{x}_t &= (1-t)\mathbf{x}_0 + t\mathbf{x}_1 \\ &= \mathbf{x}_0 + t(\mathbf{x}_1 - \mathbf{x}_0) \\ &= \mathbf{x}_0 + t\mathbf{v}\end{aligned}$$

Rearranging for $\mathbf{x}_0$:

$$\mathbf{x}_0 = \mathbf{x}_t - t\mathbf{v}$$

During inference, we do not have access to the ground truth velocity $\mathbf{v}$. Instead, we substitute the model's prediction $\mathbf{v}_\theta(\mathbf{x}_t, t) \approx \mathbf{v}$, yielding the estimated clean image:

$$\hat{\mathbf{x}}_{0,t} = \mathbf{x}_t - t\mathbf{v}_\theta(\mathbf{x}_t, t)$$

Note that this result is consistent with the n steps integration of the ODE.