TRACING COMPOSITIONAL INSTABILITY: TEMPORAL DYNAMICS OF ATTRIBUTE-OBJECT CORRESPONDENCE IN DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

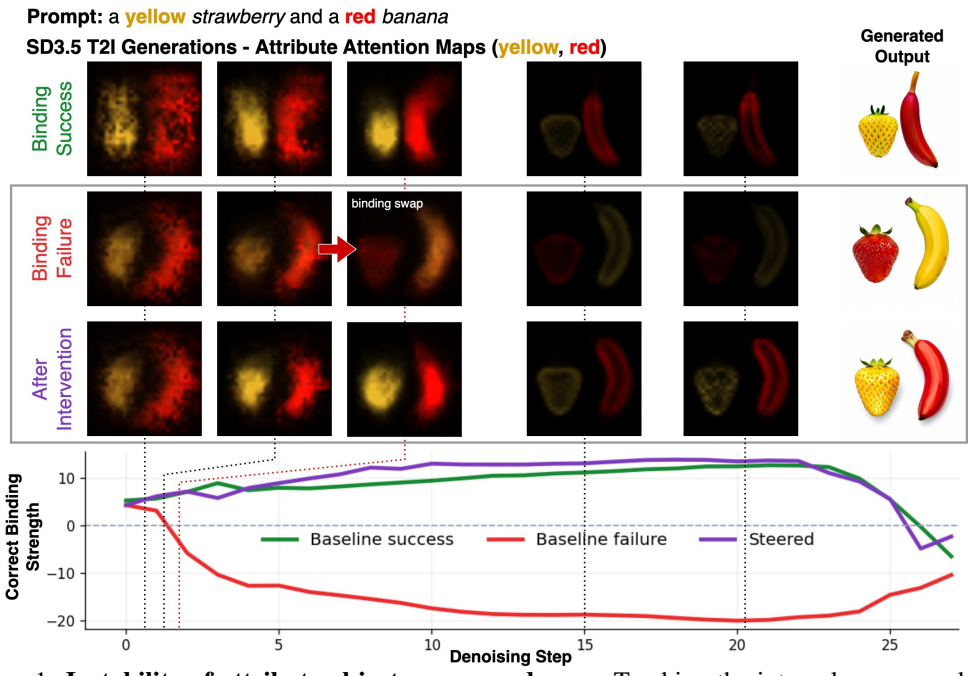

Figure 1: **Instability of attribute-object correspondences.** Tracking the internal correspondence throughout denoising reveals that failed generations can initially be correct before swapping. Our suggested inference-time steering alters the trajectory toward successful assignments.

ABSTRACT

Modern text-to-image diffusion models achieve remarkable visual fidelity, yet can fail even on simple compositions in which two objects must each realize their respective attribute. We study how such failures emerge in multimodal diffusion transformers (MMDiTs), where text and image representations are jointly updated throughout generation. Evaluation based only on the final generated image cannot reveal how these failures develop internally: *is the requested composition already incorrect from the beginning, or does it change as denoising proceeds?* This requires controlled generations in which the same semantic constituents lead to different compositional outcomes. We therefore introduce a controlled compositional benchmark that systematically varies attribute-object compositions and the conditions under which different bindings compete. Analysing token-representation geometry across blocks and denoising time, we trace how attribute-object bindings evolve internally. For the state-of-the-art MMDiT models SD3.5 and FLUX.1-dev, we find that the requested binding can be supported at earlier stages of the generative computation but become unstable as information is transformed across model depth and denoising time. Finally, we show that these internal signals are functionally actionable and introduce a test-time steering approach toward the requested binding that reduces generation failure. Together, our results characterize compositional binding not as a fixed property of the prompt representation, but as dynamically evolving throughout the generation.

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.

1 INTRODUCTION

Text-to-image generation has advanced rapidly in visual quality and prompt adherence, yet even recent models remain unreliable on surprisingly simple compositions. If asked for two objects with different attributes, a model may render correct objects and attributes but associate them incorrectly, or omit one of them altogether (Rassin et al., 2022; Conwell & Ullman, 2022; Chefer et al., 2023). A large body of work explicitly targets compositional generation failures (Feng et al., 2023; Chefer et al., 2023; Li et al., 2023; Agarwal et al., 2023; Rassin et al., 2023; Meral et al., 2024; Zhuang et al., 2024; Zarei et al., 2024; Kim et al., 2025; Lv et al., 2025; Kang et al., 2025; Li et al., 2026; Wei et al., 2026). These approaches can improve the final correspondence between prompt and image, but they are designed and evaluated at the endpoint of generation. They leave open how the internal processing of modern diffusion models gives rise to these failed compositional generations. While these benchmarks expose broad compositional limitations, their prompt sets are not designed to *systematically vary how requested attribute–object associations align or conflict with model-specific preferences*. As a result, they provide limited control over cases in which a valid but atypical request competes with a strong alternative association, for example requesting a banana with a colour different from yellow. This question is particularly pressing for multimodal diffusion transformers (MMDiTs). In architectures such as Stable Diffusion 3 and FLUX (Esser et al., 2024; Stability AI, 2024; Black Forest Labs, 2024), text is no longer a fixed conditioning signal consumed by an image denoiser. Instead, text and image tokens attend to each other, with text representations being adapted as the image evolves.

Here, we study *compositional binding*, the problem of associating each attribute with the correct entity when several entities share one representational medium (Greff et al., 2020; Campbell et al., 2024). Consider the prompt "a yellow strawberry and a red banana" (see Fig. 1). Each requested colour conflicts with the typical colour of its object. Such compositions are known to be hard because rare combinations leave a weak imprint during training (Okawa et al., 2023; Park et al., 2025; Kang et al., 2025). The set of concepts *yellow*, *strawberry*, *red*, and *banana* does not fully describe the image, but what equally important is the correspondence between them. A binding error admits two fundamentally different explanations: the requested assignment may never be supported by the model, or it may be supported earlier in the computation and become unstable as generation proceeds, which requires a mechanistic analysis of the model. Our central question is therefore: **How is a compositional assignment represented and maintained along the MMDiT generation trajectory, and where does it change when generation fails?**

To answer this, we first construct a controlled benchmark that varies how requested attribute-object assignments interact with model-specific preferences. Unlike existing benchmarks, which diagnose failures after generation, our design fixes in advance which alternative the model is drawn toward, allowing for a more systematic analysis. In particular, we follow the trajectories of image generations through the MMDiTs with a mechanistic analysis of how object representations are organized relative to the attributes in the text embedding space. Tracking these dynamics across transformer depth and denoising time in SD3.5 and FLUX.1-dev, we find that failed generations can initially exhibit an internal organization consistent with the requested assignment before progressively shifting toward the competing correspondence (Fig. 1). This complements the prompt-forgetting view of Yao et al. (2026) by showing that failures can involve a reorganization of the correspondence among prompt constituents during generation. We further propose an inference-time steering objective based on the discovered geometry that requires no masks or hand-specified attention patterns (Chefer et al., 2023; Meral et al., 2024), and is not just a static offset on the text embeddings (Zhuang et al., 2024). Our intervention is designed to mechanistically probe whether the diagnosed assignment geometry is functionally actionable, rather than serving as a general compositional correction method. Still, this steering can repair attribute swaps, together with other failure modes, such as entity/attribute neglect, in our benchmark, while reversing the steering target predominantly induces the competing crossed assignment. Importantly, these gains are achieved without degrading image fidelity.

In summary, our contributions are (1) a *compositional generation benchmark* with controlled attribute-object configurations that systematically vary alignment with model-specific preferences, (2) a *mechanistic analysis* of text-state geometry that uncovers internal changes of encoding compositionality across model depth and diffusion time in SD3.5 and FLUX.1-dev, and (3) an *inference-time steering mechanism* that improves compositional generation in MMDiTs.

2 Related Work

Compositional Text-to-Image Generation. Reliably composing multiple objects, attributes, and relations remains challenging for T2I models, shown in benchmarks such as T2I-CompBench (Huang et al., 2023), GenEval (Ghosh et al., 2023), and SANEval (Pramanik et al., 2026) which evaluate failures in attribute binding, entity generation, spatial relations, and other compositional settings. To tackle this, Structured Diffusion (Feng et al., 2023) incorporates linguistic structure into cross-attention, while Attend-and-Excite (Chefer et al., 2023), Divide-and-Bind (Li et al., 2023), and CONFORM (Meral et al., 2024) introduce attention-based objectives to improve subject preservation and attribute-object alignment. A-STAR (Agarwal et al., 2023) further highlights that semantic information present in early cross-attention may not be retained throughout denoising and explicitly regularizes this temporal consistency. Complementary approaches operate directly on the text-conditioning space: T-SAM (Kim et al., 2025) transfers syntactic structure from text self-attention into image conditioning, while (Zarei et al., 2024) identify sub-optimal CLIP conditioning as a source of attribute-binding errors. More recent work extends compositional analysis and intervention to MMDiT-based models, targeting subject mixing and neglect in architectures such as SD3 and FLUX (Wei et al., 2026). These approaches primarily diagnose or improve final compositional alignment, whereas we focus on the *temporal stability of the correspondence itself*.

Mechanistic Analysis in Diffusion Transformers. Modern MMDiTs jointly process text and image representations, making the conditioning signal part of an evolving multimodal computation rather than a strictly fixed external input (Esser et al., 2024). Recent interpretability work has shown that diffusion-transformer features contain spatially localized semantic information (Helbling et al., 2025) and that compositional structure can remain encoded even when cross-modal alignment is imperfect (Koishigarina et al., 2026). Internal text representations have further been shown to encode signals predictive of whether individual prompt concepts will be omitted, enabling targeted interventions that recover missing concepts (Baek et al., 2026). At the same time, prompt information is not necessarily preserved uniformly through network depth. Progressive degradation of linguistic information in the text branch has been documented in SD3 and FLUX, motivating explicit prompt reinjection (Yao et al., 2026). Feature-level circuit tracing has also begun to expose timestep-dependent causal pathways across text and image streams, including mechanisms related to attribute binding (Mazur et al., 2026). These results motivate treating conditioning as dynamic rather than fixed. Our focus is specifically on the stability of *structured assignments*, studying how representations connecting prompt objects and attributes reorganize over blocks and denoising time.

Temporal Dynamics in Diffusion Transformers. The denoising trajectory is inherently non-uniform, with different stages supporting different semantic and functional roles. Prior work has shown timestep-dependent behavior in editing and concept manipulation (Wu et al., 2023; Chen et al., 2024), while (Tinaz et al., 2025; Huang et al., 2026) explicitly model how internal representations change over diffusion time. Fine-grained text-image alignment is likewise not equally reliable across the trajectory. Here, CTCal (Guo et al., 2026) exploits low-noise cross-attention correspondences to calibrate noisier timesteps. Prompt-Conditioned Interventions characterize the temporal evolution of individual concepts by measuring when they can still be inserted into the generative trajectory (Görgün et al., 2026). Together, these works establish that representation, alignment, and controllability vary substantially over denoising time. Our work extends this temporal perspective from individual concepts and text-image alignment to *relational stability of prompt components*.

3 Preliminaries: Joint Attention in MMDiTs

MMDiTs replace the conventional separation between a fixed text-conditioning stream and an evolving image stream with *joint attention* over text and image tokens (Esser et al., 2024). We denote the text and image modalities by T and I, respectively. Let $\mathbf{z}_t$ denote the image latent at denoising step t, and let $\mathbf{H}_T^{l,t} \in \mathbb{R}^{n_T \times d}$ and $\mathbf{H}_I^{l,t} \in \mathbb{R}^{n_I \times d}$ denote the corresponding text and image token representations at transformer block l. Here, n_T and n_I are the numbers of text and image tokens, and d is the hidden dimension. The image-token stream is initialized from the current noisy latent $\mathbf{z}_t$ and then updated jointly with the text stream across transformer blocks.

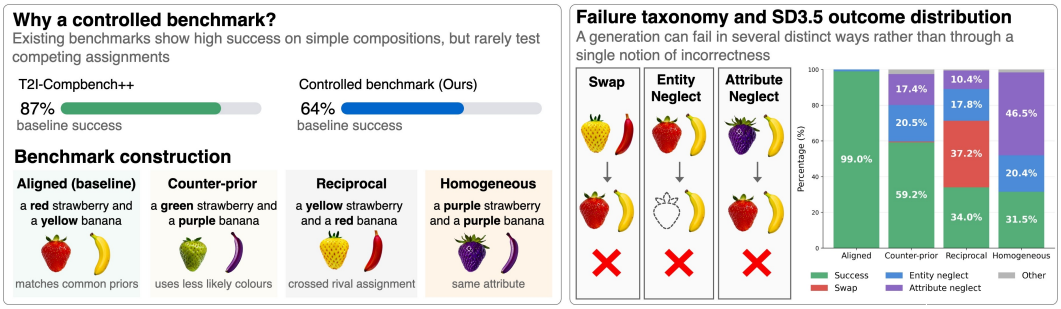

Figure 2: **Controlled benchmark design and failure taxonomy. Left:** Compositional benchmarks such as T2I-CompBench++ are largely saturated for simple compositions, whereas we deliberately vary the relationship between the requested assignment. We consider four core matched regimes: *aligned*, *counter-prior*, *reciprocal competitor*, and *homogeneous*. **Right:** We distinguish failures by their outcome, covering attribute swaps, entity neglect, and attribute neglect. Their distribution varies substantially across the four regimes: reciprocal competitors induce frequent swaps, while homogeneous prompts predominantly expose attribute neglect. This controlled variation allows us to study not only whether composition fails, but *how* the failure depends on the type of binding.

Each modality $m \in \{T, I\}$ is projected independently into queries, keys, and values,

$$\mathbf{Q}_m = \mathbf{H}_m \mathbf{W}_Q^m, \qquad \mathbf{K}_m = \mathbf{H}_m \mathbf{W}_K^m, \qquad \mathbf{V}_m = \mathbf{H}_m \mathbf{W}_V^m, \tag{1}$$

where block, denoising-step, and attention-head indices are omitted when unambiguous. The projected text and image tokens are concatenated before attention is computed,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_T \\ \mathbf{Q}_I \end{bmatrix}, \qquad \mathbf{K} = \begin{bmatrix} \mathbf{K}_T \\ \mathbf{K}_I \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \mathbf{V}_T \\ \mathbf{V}_I \end{bmatrix}. \tag{2}$$

For attention head h, with head dimension d_h, the resulting joint attention matrix is

$$\mathbf{A}^{l,t,h} = \mathrm{softmax}\left(\frac{\mathbf{Q}^{l,t,h} (\mathbf{K}^{l,t,h})^\top}{\sqrt{d_h}} \right). \tag{3}$$

Because the joint sequence contains both modalities, the attention matrix naturally decomposes into four modality-specific interaction blocks,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\mathrm{T2T}} & \mathbf{A}_{\mathrm{I2T}} \\ \mathbf{A}_{\mathrm{T2I}} & \mathbf{A}_{\mathrm{I2I}} \end{bmatrix}. \tag{4}$$

We use $\mathbf{A}_{\mathrm{X2Y}}$ to denote information flow from a source modality X to a target modality Y: the target modality supplies the queries, while the source modality supplies the keys and values. Accordingly, $\mathbf{A}_{\mathrm{T2T}}$ captures interactions among text tokens, $\mathbf{A}_{\mathrm{T2I}}$ transfers information from text tokens to image tokens, $\mathbf{A}_{\mathrm{I2T}}$ transfers information from image tokens to text tokens, and $\mathbf{A}_{\mathrm{I2I}}$ captures interactions among image tokens. This bidirectional coupling distinguishes MMDiTs from conventional cross-attention-based diffusion architectures, where encoded text typically serves as a fixed conditioning signal while the image representation evolves during denoising. In MMDiTs, the external text-encoder output remains fixed, but the internal text representation is updated through joint attention with the evolving image stream. Consequently, the text representation at a given transformer block can depend on the denoising state, i.e. $\mathbf{H}_T^{l,t} \neq \mathbf{H}_T^{l,t'}, t \neq t'$. This gives rise to two complementary dimensions of variation: **network depth** l, corresponding to transformations across successive transformer blocks, and **denoising time** t, corresponding to changes induced by the evolving image state. Throughout the paper, we use this block-timestep view to trace how the organization of prompt constituents changes during generation.

4 Isolating Relational Failure: A Controlled Benchmark

The joint processing of text and image representations motivates studying whether the model preserves the requested correspondence between attributes and objects, which we refer to as **attribute-object binding**. A generation is **binding-correct** when the realized correspondences match the

prompt. This distinguishes relational errors from failures such as entity or attribute neglect; in our setting, the clearest binding failure is a complete exchange of the two attributes between the two objects. A failed multi-object generation, however, does not necessarily imply a relational failure. A requested object-attribute pair may already be difficult to realize in isolation, or the request may compete with a strong learned object-attribute preference. We therefore construct a controlled benchmark that explicitly accounts for these factors before introducing composition. The estimation of model-specific preferences, single-object screening, and complete benchmark construction are described in App. Sec. A.

4.1 Benchmark Design

Controlled setting. We consider prompts of the form "a A_1N_1 and a A_2N_2", where A_i denotes an attribute and N_i an object. The prompt specifies the requested assignment $\{A_1 \leftrightarrow N_1,\ A_2 \leftrightarrow N_2\}$. When $A_1 \neq A_2$, the same constituent concepts also admit a complete crossed assignment, $\{A_1 \leftrightarrow N_2,\ A_2 \leftrightarrow N_1\}$. For example, the requested composition "a green strawberry and a red broccoli" has the crossed alternative "a red strawberry and a green broccoli." The two outcomes contain exactly the same objects and attributes but differ in their correspondence. We use this minimal prompt structure to keep syntax and scene complexity fixed while allowing the same constituents to be tracked consistently through the model, with more discussion in App. Sec. A.1.

Controlling non-compositional difficulty. To construct typical and atypical compositions, we estimate for each object N a model-specific preferred attribute $c^*(N)$ from neutral generations, likely corresponding to training data priors, and measure whether each requested object-attribute pair can be reliably realized when generated in isolation. This screening controls for failures caused by intrinsic single-object difficulty independent of compositional problems. We retain only object-attribute pairs that pass this single-object screen when constructing the Controlled Benchmark, reducing failures attributable to constituents that are already unreliable in isolation. The estimation procedure and filtering criteria, and single-object results can be found in App. Secs. A.2 and A.3.

Matched compositional conditions. Through model-specific preferences $c^*(N)$, we construct four core matched conditions that vary the relationship between objects and attributes. In the *aligned* condition, each object receives its preferred attribute. In the *counter-prior* condition, both requested associations oppose these preferences. In the *reciprocal competitor* condition, assignments are swapped, i.e. $A_1 = c^*(N_2), A_2 = c^*(N_1)$. Finally, in the *homogeneous* condition, $A_1 = A_2$, so exchanging the attributes leaves the requested composition unchanged and no distinct crossed assignment exists. We provide additional details in App. Sec. A.4.

For each prompt and condition we generate 100 samples using a fixed set of random seeds. We additionally perform a matched-initial-noise analysis in which the same seed is evaluated across different compositional conditions for a given object pair. This keeps the initial latent fixed while changing the assignment, allowing us to test whether compositional success is determined by the initialization. We observe that the same initial noise can yield different outcomes under different assignments, indicating that it influences the generation trajectory but does not uniquely determine its compositional outcome. We visualize these seed-matched outcomes and provide further details in App. Sec. A.5.

4.2 Evaluation and Failure Diagnosis

Following the attribute-binding evaluation of T2I-CompBench (Huang et al., 2023), we evaluate the two requested object-attribute bindings independently. For a generated image I, the evaluator estimates whether A_1N_1 and A_2N_2 are each present in the complete image. Replacing BLIP-VQA with Qwen3-8B, we retain the original multiplicative formulation of binding score $S_{\text{bind}}(I)$,

$$S_{\text{bind}}(I) = p(A_1N_1 \mid I)\, p(A_2N_2 \mid I). \tag{5}$$

Importantly, this score measures the presence of the composition, but does not identify how generation fails. We therefore introduce diagnostic queries for object presence, attribute scorability, and, when defined, crossed attribute assignment. Each image is assigned either *success*, *entity neglect*, *attribute swap*, *attribute neglect*, or OTHER. OTHER comprises cases in which an object's attribute cannot be scored due to ambiguity, occlusion, or fusion, and generations that satisfy neither the requested assignment nor a crossed assignment. The VQA protocol is provided in App. Sec. A.6.

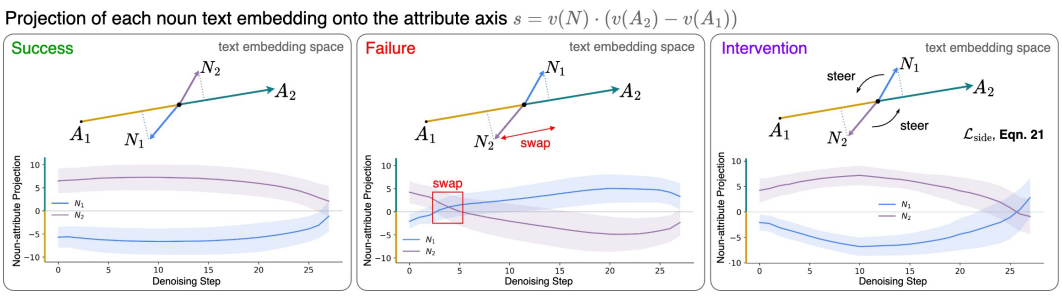

Figure 3: **Tracking and steering attribute-object assignment geometry.** We project each noun representation onto the vector defined by the two attribute representations (Eq. (8)) and trace projections during denoising for Block-12 of SD3.5. **Left:** Successful generations preserve the requested ordering, with projected N_1 closer to A_1 and N_2 closer to A_2. **Middle:** In failed generations, this assignment can initially be consistent with the prompt but later reverse, yielding a crossed attribute-object correspondence. **Right:** Our suggested steering uses the model's early assignment geometry as a target and penalizes deviations from the requested sides of the attribute axis, steering the trajectory toward the successful configuration. Shaded regions show variance across generations.

5 Tracing Compositional Assignments Through Generation

To understand when a compositional failure occurs, we trace generations across transformer depth and denoising time. We specifically measure how noun representations are organized relative to their attributes and characterize the depth-dependent evolution of the text representations. We refer to App. Sec. B for complete block-wise analyses.

5.1 Measuring the Evolving Assignment

For $X \in \{A_1, N_1, A_2, N_2\}$, let $\mathcal{S}_X$ denote its text-token positions. We represent each X at the input to transformer block l and denoising step t by averaging the corresponding rows of the text hidden state $\mathbf{H}_T^{l,t}$

$$\mathbf{x}_X^{l,t} = \frac{1}{|\mathcal{S}_X|} \sum_{i \in \mathcal{S}_X} \mathbf{H}_T^{l,t}[i,:]. \tag{6}$$

These representations are measured before block-specific query, key, and value projections.

Assignment geometry. We first define a normalized attribute direction from A_1 to A_2,

$$\mathbf{d}_A^{l,t} = \frac{\mathbf{x}_{A_2}^{l,t} - \mathbf{x}_{A_1}^{l,t}}{\left\| \mathbf{x}_{A_2}^{l,t} - \mathbf{x}_{A_1}^{l,t} \right\|_2}. \tag{7}$$

Centering the axis between the two attributes and projecting each noun onto it gives

$$s_i^{l,t} = \left\langle \mathbf{x}_{N_i}^{l,t} - \frac{1}{2}\left(\mathbf{x}_{A_1}^{l,t} + \mathbf{x}_{A_2}^{l,t}\right), \mathbf{d}_A^{l,t} \right\rangle, \quad i \in \{1,2\}. \tag{8}$$

In the correct scenario, projected N_1 is placed toward A_1 and N_2 toward A_2, yielding $s_1^{l,t} < 0 < s_2^{l,t}$, describing the organization of the internal text state.

Depth-dependent text update. We further characterize how the representations change relative to the state at the input of the transformer. For each X, we define its displacement and remove the component shared across all four as

$$\mathbf{r}_X^{l,t} = \mathbf{x}_X^{l,t} - \mathbf{x}_X^{0,t}, \qquad \tilde{\mathbf{r}}_X^{l,t} = \mathbf{r}_X^{l,t} - \frac{1}{|\mathcal{C}|} \sum_{Y \in \mathcal{C}} \mathbf{r}_Y^{l,t}. \tag{9}$$

For attribute A_i and noun N_j, let $\gamma_{ij}^{l,t} = \cos\left(\tilde{\mathbf{r}}_{A_i}^{l,t}, \tilde{\mathbf{r}}_{N_j}^{l,t}\right)$ denote the similarity of their centered updates. We summarize the relative preference for requested versus crossed pairs as

$$M_{l,t} = \frac{1}{2}\left[\left(\gamma_{11}^{l,t} - \gamma_{21}^{l,t}\right) + \left(\gamma_{22}^{l,t} - \gamma_{12}^{l,t}\right)\right], \quad l > 0. \tag{10}$$

The sign of $M_{l,t}$ indicates whether the update is more consistent with the requested or crossed pairs, its magnitude reflects the strength of preference.

5.2 Assignment Dynamics Across Outcomes

We analyze each failure category separately, retaining prompts that produce at least one seed with the corresponding outcome. When comparing failed and successful trajectories, successful seeds are drawn from the same prompts whenever available. This reduces differences due to prompt semantics and focuses the comparison on outcome-dependent trajectory dynamics (see App. Sec. B.1).

Unstable assignments during generation. Successful generations tend to maintain the expected noun projections, with N_1 toward A_1 and N_2 toward A_2 across the generation trajectory. Intriguingly, complete attribute swaps are *not always organized according to the crossed assignment* from the start. In failed generations, the noun representations can initially *support the requested assignment* before progressively shifting toward the competing attributes at later blocks and denoising steps (Fig. 3, Sec. B.2). A wrong output can therefore arise from a reorganization of the requested correspondence during generation rather than from an assignment that was already incorrect at initialization. The two noun trajectories are particularly informative when considered separately. A change in correspondence need not involve symmetric movement of both nouns, with one representation moving strongly toward its competing attribute while the other remains comparatively stable.

Failure modes show different internal structure. In contrast to attribute swaps, in neglect failures, the relative attribute-noun structure can become weak or ambiguous. This distinction is reflected in the depth-dependent update $M_{l,t}$ showing not only which pairing is favored, but also how strongly the competing assignments are differentiated. These differences become visible early in the generation, making the text update useful for early identification of generations at risk of failure.

6 Mechanistically Grounded Intervention for Compositionality

Our trajectory analysis suggests that compositional failures are often preceded by an early internal change of the attribute-object assignment. In the following we show that this signal can be used as a steering mechanism using a selective *scout-and-replay* strategy. We first run a short, unmodified prefix of the denoising trajectory to estimate whether the current prompt-seed pair is at risk of compositional failure. The trajectories identified as risky are replayed with steering, hence this mechanism is inference-time, while the detector itself is a fixed threshold rule calibrated from baseline trajectories rather than a learned classifier. The intervention only attempts to preserve the assignment geometry already expressed earlier in the trajectory. The overall formulation is shared across backbones, while the internal states used for detection, steering schedule, and optimization hyperparameters are model-specific. We report these implementation details in App. Sec. C.

6.1 Early Failure Signal

We first start the denoising trajectory and evaluate the text-update margin $M_{l,t}$ from Eq. (10) at a small set of early internal states. A strong positive margin indicates support for the requested pairing, a reduced margin indicates that this support has weakened or shifted toward the competing correspondence. The selected measurements are aggregated into a backbone-specific failure-risk score R. We intervene when this score crosses a fixed threshold τ. Construction of R, the internal states used, and the threshold calibration are reported in App. Sec. C.1. After the intervention decision is made, we replay the trajectory from the saved pre-scout state and apply the intervention described below. In App. Sec. C.4, we ablate on applying steering to every trajectory.

6.2 Latent steering

For a trajectory selected for intervention, we use the attribute-axis geometry introduced in Sec. 5.1 as steering signal. At denoising step t_a, we retain internal states for which the requested assignment is expressed, i.e. $s_1^{l,t_a} < 0 < s_2^{l,t_a}$ and define the corresponding assignment separation as

$$D_l^{\text{anchor}} = s_2^{l,t_a} - s_1^{l,t_a}. \tag{11}$$

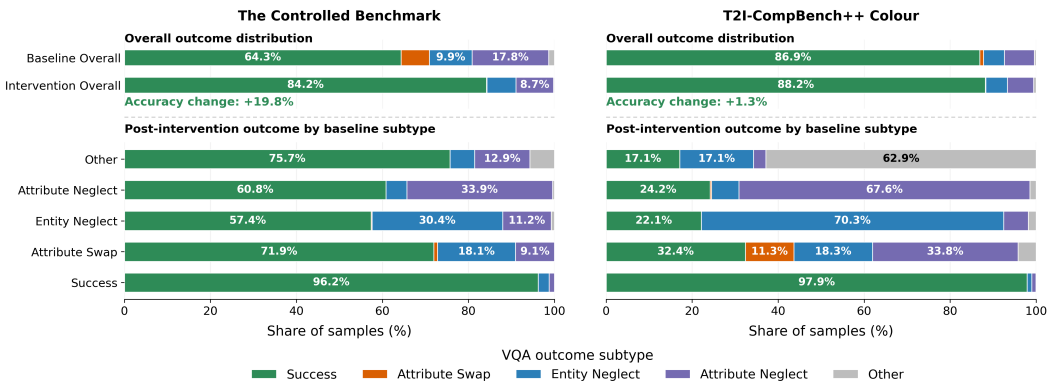

Figure 4: **Intervention effects on SD3.5. Top:** Compositional success before and after intervention on our Controlled Benchmark and T2I-CompBench++ Colour. **Bottom:** Post-intervention outcomes stratified by VQA label on success and failure. Each row contains only samples assigned to the corresponding category before intervention, colored segments indicate outcome after intervention. The intervention produces substantially stronger corrections on the Controlled Benchmark while largely preserving baseline successes on both benchmarks.

For each retained state, we define a target side margin $m_l = \eta D_l^{\text{anchor}}$, where η determines how strongly the anchor geometry is preserved. During subsequent guided denoising, we recompute the attribute axis and noun projections from the current text representations and minimize

$$\mathcal{L}_{\text{side}}^{l,t} = \left[m_l + s_1^{l,t}\right]_+ + \left[m_l - s_2^{l,t}\right]_+, \qquad \text{with } [x]_+ = \max(0, x). \tag{12}$$

The objective encourages N_1 and N_2 to remain on their respective sides of the evolving A_1–A_2 axis. We also do not directly overwrite the text representations. Because the text and image streams interact through joint attention, $\mathcal{L}_{\text{side}}^{l,t}$ is backpropagated through the current image latent $\mathbf{z}_t$

$$\mathbf{z}_t \leftarrow \mathbf{z}_t - \alpha \frac{\nabla_{\mathbf{z}_t} \mathcal{L}_{\text{side}}^{l,t}}{\left\| \nabla_{\mathbf{z}_t} \mathcal{L}_{\text{side}}^{l,t} \right\|_2}, \tag{13}$$

before continuing the standard denoising update. The intervention is therefore self-anchored in two senses: it derives the desired binding configuration from the model itself, and it acts through the existing image-to-text feedback pathway rather than replacing the text conditioning with an externally constructed representation.

6.3 Intervention Results

We evaluate the intervention on both our Controlled Benchmark introduced in Sec. 4 and the colour subset of T2I-CompBench++ (Huang et al., 2025). We report the full SD3.5 results in Fig. 4, including transitions between VQA failure categories. FLUX.1-dev results and the corresponding transition analysis are reported in App. Sec. C.3. In App. Sec. C.4, we compare the intervention with an always-on variant that applies the same steering to every trajectory, thereby isolating the effect of failure-based selection. As a directional control, we additionally reverse the steering target on baseline-successful generations: this predominantly induces the competing crossed assignment, whereas random-direction perturbations cause substantially more diffuse failures (App. Sec. C.5).

Overall effectiveness. On SD3.5, the intervention increases success on the Controlled Benchmark from 64.3% to 84.2%, an improvement of 19.8% . Equivalently, the aggregate failure rate decreases from 35.7% to 15.8%, removing approximately 56% of the baseline errors. The improvement is smaller but remains positive on T2I-CompBench++ Colour, where the success rate increases from 86.9% to 88.2% ($+1.3\%$). This benchmark already has a substantially higher baseline success rate, leaving considerably fewer trajectories available for correction. Importantly, the gain on failing trajectories is not obtained by broadly destabilizing successful generations. Among samples that are successful at baseline, 96.2% remain successful after intervention on the Controlled Benchmark,

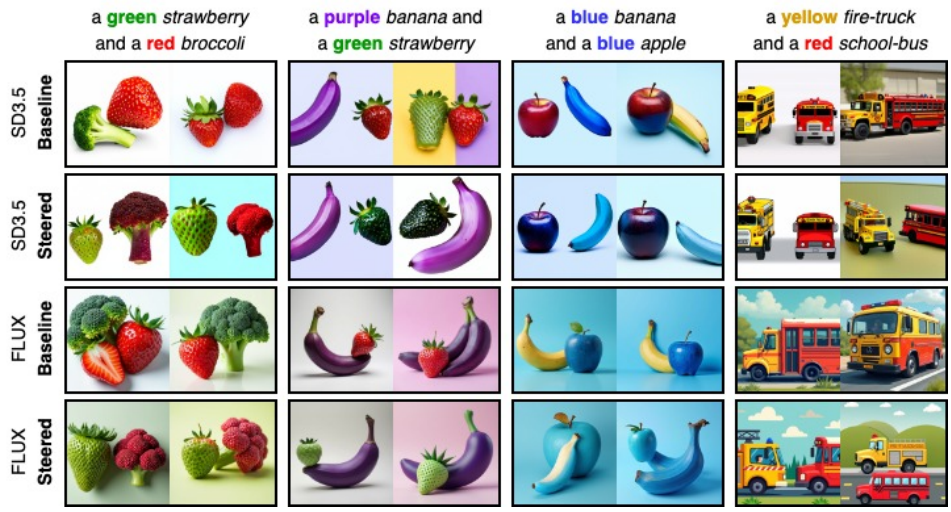

Figure 5: **Qualitative effect of self-anchored steering.** Each column shows generations from the same prompt before and after intervention. Baseline samples exhibit compositional errors including attribute swaps, attribute neglect, and incorrect object attribute associations. After steering, the requested attribute-object correspondences are recovered more consistently.

and 97.9% remain successful on T2I-CompBench++ Colour. This high retention rate is a direct benefit of selectively intervening only on trajectories identified as at risk. Qualitative examples in Fig. 5 and App. Sec. C.6 illustrate these effects where the intervention can recover the requested attribute-object assignments while largely preserving object identity and overall scene structure.

Image quality. We additionally verify that the intervention does not trade compositional correctness for visual quality using FID (Heusel et al., 2017), which is computed in Inception (Szegedy et al., 2016) feature space against ImageNet (Deng et al., 2009) reference images. As shown in Tab. 1, image quality is preserved and slightly improves after intervention for both SD3.5 and FLUX.1-dev. This indicates that the observed gains in compositional correctness are not achieved at the expense of overall quality.

Benchmark	Model	Base	Ours
Controlled	SD3.5	165.5	**158.9**
	FLUX.1-dev	161.9	**153.0**
T2I-CompBench++	SD3.5	74.3	**73.6**
	FLUX.1-dev	75.7	**75.4**

Table 1: **Image quality.** FID (↓) before and after intervention.

Which failures are repaired? On our Controlled Benchmark, the intervention converts 71.9% of attribute-swap failures into successes, together with 60.8% of attribute-neglect failures and 57.4% of entity-neglect failures. Even among failures assigned to the heterogeneous OTHER category, 75.7% become successful. This highlights that the intervention effect is not restricted to attribute swaps. The transition matrix also exposes that not every corrected assignment produces a fully successful image. For example, among baseline attribute-swap failures on the Controlled Benchmark, 18.1% become entity-neglect cases and 9.1% become attribute-neglect cases rather than successes. Yet, overall we achieve much higher success rates across cases than without an intervention.

7 CONCLUSIONS

We study compositional binding in MMDiT-based text-to-image models by tracing attribute-object correspondences across transformer depth and denoising time. Our results show that failures can emerge from an initially correct internal organization that becomes unstable during generation. Building on this signal, we introduce a selective self-anchored intervention that improves compositional generation by preserving the earlier requested assignment. Together, these findings characterize binding as a dynamic property of the generative trajectory. We further discuss limitations of our method in App. Sec. D.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with writing, including language editing, rewriting, and improving the clarity and organization of the manuscript. We have not used generative AI tools to generate or alter experimental results, and other required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for occasional assistance with code development, debugging, and the presentation of research ideas. We have reviewed all AI-assisted work. AI-assisted text was reviewed and edited by the authors, and AI-assisted code was verified and tested for correctness before being used in our experiments. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To make our work reproducible, we provide details on Experimental Setup and Method Implementation in App. Secs. A to C. We will further make code and benchmark datasets publicly available after acceptance.

REFERENCES

Aishwarya Agarwal, Srikrishna Karanam, K J Joseph, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. A-STAR: Test-time attention segregation and retention for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2283–2293, 2023.

Kanghyun Baek, Jaihyun Lew, Chaehun Shin, Jungbeom Lee, and Sungroh Yoon. Diagnosing and correcting concept omission in multimodal diffusion transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M. Frankland, Thomas L. Griffiths, Jonathan D. Cohen, and Taylor Webb. Understanding the limits of vision language models through the lens of the binding problem. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-Excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics*, 42(4):148:1–148:10, 2023. doi: 10.1145/3592116.

Sherry X Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. TiNO-Edit: Timestep and noise optimization for robust diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6337–6346, 2024.

Colin Conwell and Tomer Ullman. Testing relational understanding in text-guided image generation. *arXiv preprint arXiv:2208.00005*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *International Conference on Learning Representations*, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ada Görgün, Fawaz Sammani, Nikos Deligiannis, Bernt Schiele, and Jonas Fischer. Temporal concept dynamics in diffusion models via prompt-conditioned interventions. In *International Conference on Learning Representations*, 2026.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9380–9389, 2024.

Xiefan Guo, Xinzhu Ma, Haiyu Zhang, and Di Huang. CTCal: Rethinking text-to-image diffusion models via cross-timestep self-calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43558–43567, 2026.

Alec Helbling, Tuna Han Salih Meral, Benjamin Hoover, Pinar Yanardag, and Duen Horng Chau. ConceptAttention: Diffusion transformers learn highly interpretable features. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22946–22963, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025. doi: 10.1109/TPAMI.2025.3531907.

Victor Shea-Jay Huang, Le Zhuo, Yi Xin, Zhaokai Wang, Fu-Yun Wang, Yuchi Wang, Renrui Zhang, Peng Gao, and Hongsheng Li. TIDE: Temporal-aware sparse autoencoders for interpretable diffusion transformers in image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 435–443, 2026. doi: 10.1609/aaai.v40i1.37006.

Seil Kang, Woojung Han, Dayun Ju, and Seong Jae Hwang. Rare text semantics were always there in your diffusion transformer. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jeeyung Kim, Erfan Esmaeili, and Qiang Qiu. Text embedding is not all you need: Attention control for text-to-image semantic alignment with text self-attention maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8031–8040, 2025.

Darina Koishigarina, Arnas Uselis, and Seong Joon Oh. CLIP behaves like a bag-of-words model cross-modally but not uni-modally. In *International Conference on Learning Representations*, 2026.

Binglei Li, Mengping Yang, Zhiyu Tan, Junping Zhang, and Hao Li. Unraveling MMDiT blocks: Training-free analysis and enhancement of text-conditioned diffusion. *arXiv preprint arXiv:2601.02211*, 2026.

Yumeng Li, Margret Keuper, Dan Zhang, and Anna Khoreva. Divide & bind your attention for improved generative semantic nursing. In *34th British Machine Vision Conference (BMVC 2023)*. BMVA Press, 2023.

Zhengyao Lv, Tianlin Pan, Chenyang Si, Zhaoxi Chen, Wangmeng Zuo, Ziwei Liu, and Kwan-Yee K. Wong. Rethinking cross-modal interaction in multimodal diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5934–5943, 2025.

Artyom Mazur, Nina Konovalova, and Aibek Alanov. DifFRACT: Diffusion feature reconstruction and attribution for circuit tracing. *arXiv preprint arXiv:2606.15796*, 2026.

Tuna Han Salih Meral, Enis Simsar, Federico Tombari, and Pinar Yanardag. CONFORM: Contrast is all you need for high-fidelity text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9005–9014, 2024.

Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Dongmin Park, Sebin Kim, Taehong Moon, Minkyu Kim, Kangwook Lee, and Jaewoong Cho. Rare-to-frequent: Unlocking compositional generation power of diffusion models on rare concepts with LLM guidance. In *International Conference on Learning Representations*, 2025.

Rishav Pramanik, Ian E. Nielsen, Jeff Smith, Saurav Pandit, Ravi P. Ramachandran, and Zhaozheng Yin. SANEval: Open-vocabulary compositional benchmarks with failure-mode diagnosis. *arXiv preprint arXiv:2602.00249*, 2026.

Royi Rassin, Shauli Ravfogel, and Yoav Goldberg. DALL-E 2 is seeing double: Flaws in word-to-concept mapping in Text2Image models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 335–345, 2022. doi: 10.18653/v1/2022.blackboxnlp-1.28.

Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Stability AI. Stable Diffusion 3.5. `https://github.com/Stability-AI/sd3.5`, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Tianyi Wei, Dongdong Chen, Yifan Zhou, and Xingang Pan. Enhancing MMDiT-based text-to-image models for similar subject generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):7692–7706, 2026. doi: 10.1109/TPAMI.2026.3663759.

Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910, 2023.

Yuxuan Yao, Yuxuan Chen, Hui Li, Kaihui Cheng, Qipeng Guo, Yuwei Sun, Zilong Dong, Jingdong Wang, and Siyu Zhu. Prompt reinjection: Alleviating prompt forgetting in multimodal diffusion transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Arman Zarei, Keivan Rezaei, Samyadeep Basu, Mehrdad Saberi, Mazda Moayeri, Priyatham Kattakinda, and Soheil Feizi. Improving compositional attribute binding in text-to-image generative models via enhanced text embeddings. *arXiv preprint arXiv:2406.07844*, 2024.

Chenyi Zhuang, Ying Hu, and Pan Gao. Magnet: We never know how text-to-image diffusion models work, until we learn how vision-language models function. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

APPENDIX

TABLE OF CONTENTS

(A) Controlled Benchmark Construction and Evaluation **14**

A.1 Minimal Two-Object Setting 14

A.2 Estimating Model-Specific Attribute Preferences 14

A.3 Single-Object Executability 16

A.4 Matched Compositional Conditions 17

A.5 Matched Initial Noise 18

A.6 VQA Evaluation and Failure Taxonomy 19

(B) Additional Analysis of Compositional Assignment Dynamics **19**

B.1 Trajectory Grouping and Analysis Protocol 19

B.2 Assignment Geometry Across Blocks and Denoising Time 20

B.3 Depth-Dependent Text Updates 20

(C) Intervention Details and Additional Results **37**

C.1 Early Failure Detection 37

C.2 Model-Specific Intervention Configuration 37

C.3 Additional FLUX.1-dev Results 38

C.4 Effect of Selective Intervention 38

C.5 Directional Intervention Controls 40

C.6 Additional Qualitative Results 41

(D) Limitations **43**

A Controlled Benchmark Construction and Evaluation

This section provides the complete construction and evaluation protocol for the controlled benchmark introduced in Sec. 4. The benchmark is designed to isolate failures that arise specifically under composition from two important confounders: a requested object-attribute pair may already be difficult to generate in isolation, and the requested association may compete with a strong object-specific preference of the generative model. We therefore construct the benchmark in three stages. We first estimate model-specific object-attribute preferences from neutral generations (Sec. A.2). We then measure whether candidate object-attribute pairs can be realized reliably in isolation (Sec. A.3). Finally, we combine the screened pairs into matched compositional conditions that systematically vary the relationship between the requested assignment and the model's learned preferences (Sec. A.4). We additionally perform a matched-initial-noise analysis to test whether compositional success or failure is determined by the stochastic initialization, as described in Sec. A.5.

All benchmark construction steps are performed independently for each generative backbone. Thus, neither the preferred attribute of an object nor the resulting candidate set is assumed to be identical across models.

Generation setup. Unless otherwise stated, all images are generated at 512×512 resolution using 28 inference steps with the `FlowMatchEulerDiscreteScheduler`. For SD3.5, we use `stabilityai/stable-diffusion-3.5-medium` with classifier-free guidance scale 7.0 and FP16 inference. For FLUX.1-dev, we use `black-forest-labs/FLUX.1-dev` with its embedded guidance scale 3.5 and BF16 inference. We use an empty negative prompt for SD3.5; FLUX.1-dev is run without true classifier-free guidance.

A.1 Minimal Two-Object Setting

Our goal is to study failures of attribute-object assignment rather than failures caused by complex syntax, counting, spatial relations, or long prompts. We therefore use the minimal template

$$\text{“a } A_1N_1 \text{ and a } A_2N_2\text{”}, \tag{A.1}$$

where A_i denotes an attribute and N_i an object.

This construction keeps the syntactic structure fixed while allowing the relationship between the same constituent concepts to be changed systematically. In particular, when $A_1 \neq A_2$, the requested and crossed assignments contain exactly the same two objects and two attributes and differ only in their correspondence. This makes the setting suitable both for controlled behavioral evaluation and for tracking the same constituents through the transformer in Sec. 5.

The simplicity of the template is therefore deliberate: the benchmark is not intended to measure general prompt complexity, but to provide a diagnostic setting in which the requested correspondence can be manipulated while other sources of complexity remain fixed. Additionally, to reduce dependence on a particular ordering of the two constituents, we also consider both prompt orders, “a A_1N_1 and a A_2N_2” and “a A_2N_2 and a A_1N_1”, while preserving the same semantic attribute-object assignment.

A.2 Estimating Model-Specific Attribute Preferences

A requested association may be difficult because it conflicts with an object-attribute preference already encoded by the generative model. We therefore estimate these preferences directly from generations rather than assuming human-defined canonical attributes.

For each object N, we generate images from neutral prompts that do not specify a target attribute. To reduce dependence on a particular prompt surface form, we use multiple neutral templates,

$$\begin{aligned} &\text{“a } N\text{”},\\ &\text{“a photo of a } N\text{”},\\ &\text{“a } N \text{ on a table”}. \end{aligned} \tag{A.2}$$

For each neutral template, we generate 100 samples using the same set of seeds across templates whenever possible.

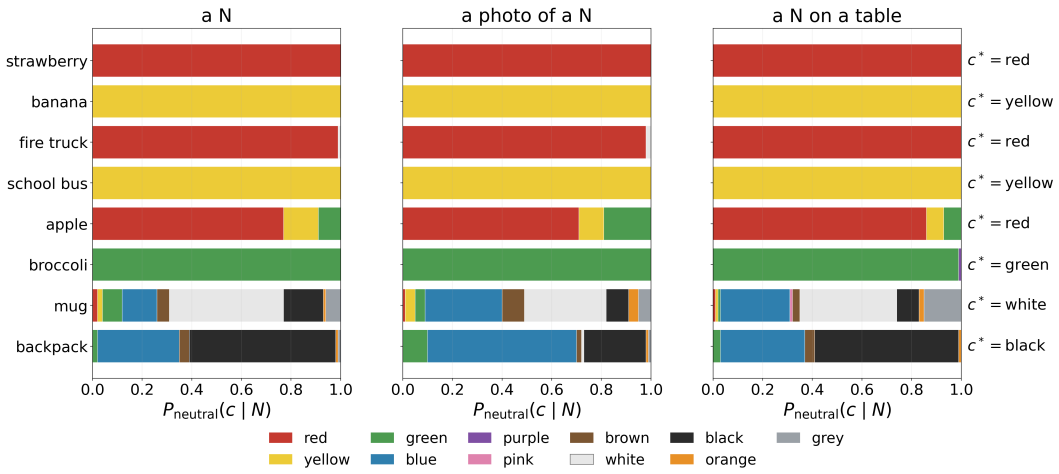

Figure A.1: **Model-specific attribute preferences for SD3.5.** Empirical attribute distributions obtained from neutral generations for each object, shown separately for the three neutral prompt templates. The model-preferred attribute $c^*(N)$ is defined as the most frequently observed attribute across valid neutral generations.

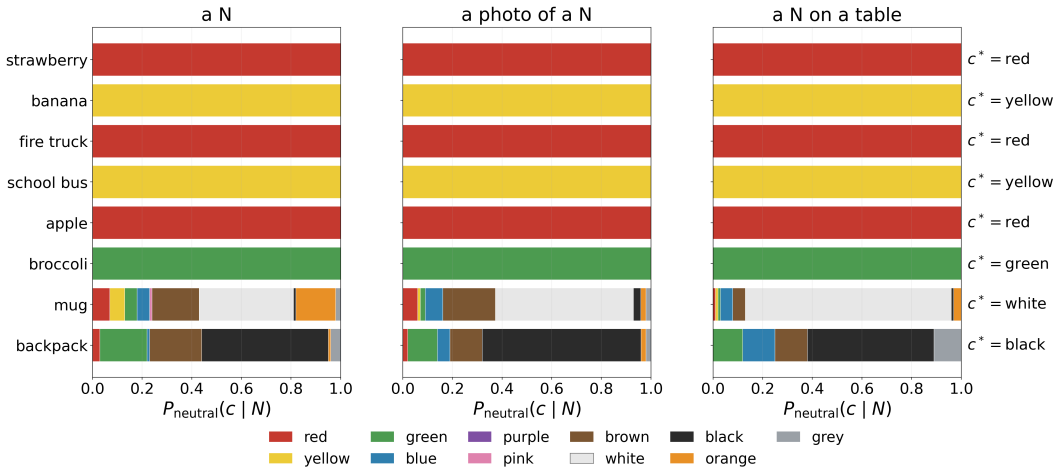

Figure A.2: **Model-specific attribute preferences for FLUX.1-dev.** Empirical attribute distributions obtained from neutral generations for each object, shown separately for the three neutral prompt templates. The model-preferred attribute $c^*(N)$ is defined as the most frequently observed attribute across valid neutral generations.

Each generation is evaluated in three stages. We first determine whether the object is present, then whether its identity is recognizable independently of its attribute, and finally whether the attribute is visually scorable. Only generations satisfying all three conditions contribute to the empirical attribute distribution.

Let $\mathcal{I}_N^{\text{neutral}}$ denote the set of neutral generations in which N is present, recognizable, and attribute-scorable. For an attribute c, we compute

$$P_{\text{neutral}}(c \mid N) = \frac{\sum_{I \in \mathcal{I}_N^{\text{neutral}}} \mathbf{1}\{\widehat{c}(I, N) = c\}}{|\mathcal{I}_N^{\text{neutral}}|}.$$

where $\widehat{c}(I, N)$ denotes the observed attribute assigned by the evaluator. The distribution $P_{\text{neutral}}(c \mid N)$ is computed by pooling all valid generations across the three neutral prompt templates. The model-specific preferred attribute is then

$$c^*(N) = \arg\max_c P_{\text{neutral}}(c \mid N).$$

Thus, $c^*(N)$ denotes the attribute that the model most frequently realizes for N under neutral prompting, rather than a manually specified canonical attribute. The resulting distributions are shown in Figs. A.1 and A.2 for SD3.5 and FLUX.1-dev, respectively, where we report the estimates separately across neutral templates to verify whether the selected preference is stable with respect to prompt phrasing.

A.3 Single-Object Executability

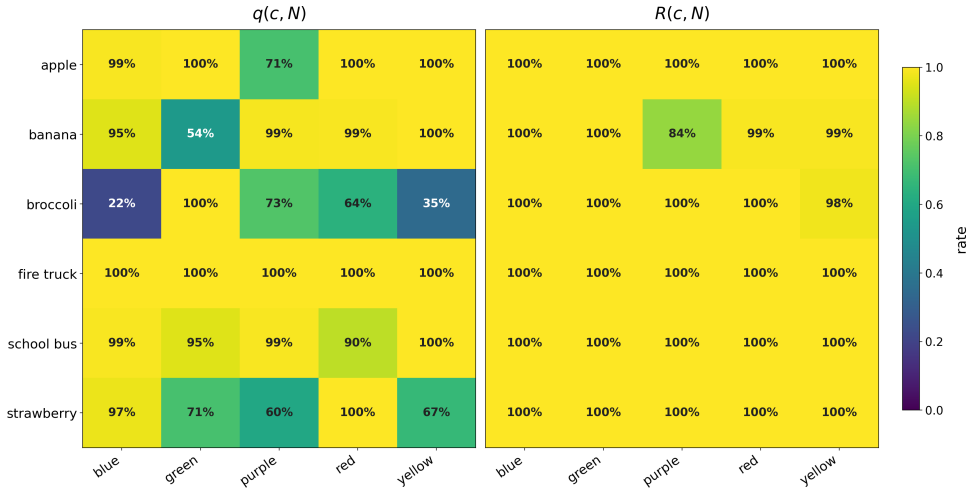

Figure A.3: **Single-object executability for SD3.5.** For each object-attribute pair, we report the conditional attribute realization rate $q(c, N)$ (left) and object identity preservation $R(c, N)$ (right). These measurements are used to screen pairs before constructing the Controlled Benchmark.

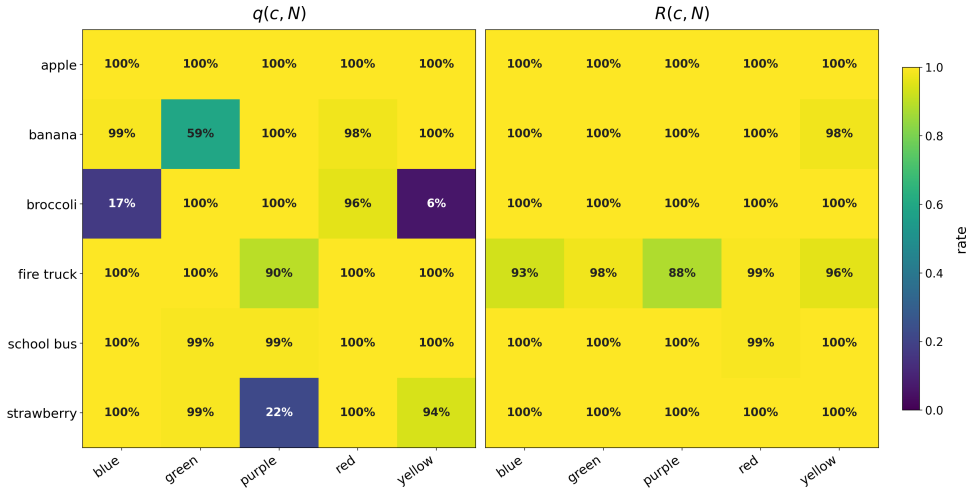

Figure A.4: **Single-object executability for FLUX.1-dev.** For each object-attribute pair, we report the conditional attribute realization rate $q(c, N)$ (left) and object identity preservation $R(c, N)$ (right). These measurements are used to screen pairs before constructing the Controlled Benchmark.

A failure under composition is difficult to interpret if one of the requested object-attribute pairs is already unreliable when generated alone. We therefore evaluate every candidate pair independently before constructing two-object prompts. To match the local structure of the compositional template

"a A_1N_1 and a A_2N_2", each constituent pair is evaluated using the corresponding single-object prompt "a A_iN_i". For each candidate object-attribute pair, we generate 100 samples using the same fixed set of seeds used throughout the benchmark whenever possible.

For an object N and requested attribute c, let $\mathcal{I}_{N,c}^{\text{single}}$ denote the set of single-object generations in which N is present, recognizable, and attribute-scorable. We define the conditional single-object realization rate as

$$q(c, N) = \frac{\sum_{I \in \mathcal{I}_{N,c}^{\text{single}}} \mathbf{1}\{\widehat{c}(I, N) = c\}}{|\mathcal{I}_{N,c}^{\text{single}}|}.$$

This quantity measures whether the requested attribute can be realized once the target object itself has been generated successfully.

We separately record object identity preservation,

$$R(c, N) = \frac{\#\{\text{generations in which } N \text{ is recognizable}\}}{\#\{\text{generated images}\}}, \tag{A.3}$$

which allows failures of object identity to be distinguished from failures of attribute realization.

The resulting single-object measurements for SD3.5 and FLUX.1-dev are summarized in Figs. A.3 and A.4, respectively. We require each requested object-attribute pair to have a single-object realization rate $q(c, N) > 0.5$, such that the requested attribute is realized more often than not when the object is generated in isolation. Object identity preservation $R(c, N)$ is reported separately to distinguish failures of recolouring from failures of object identity. All object-attribute pairs used in the Controlled Benchmark satisfy this capability criterion; the lowest retained single-object realization rate is 0.59.

A.4 Matched Compositional Conditions

Using the object-attribute pairs retained by the single-object executability screen and the model-specific preferences $c^*(N)$, we construct four core compositional regimes. Restricting the benchmark to pairs that are individually realizable reduces the possibility that failures in the two-object setting are driven simply by an inability to generate one of the requested constituents in isolation.

All conditions use the same prompt template from Eq. (A.1); only the relationship between the requested assignment and the model's estimated preferences is changed. The four regimes and their empirical failure distributions are illustrated in Fig. 2.

Aligned. The requested assignment agrees with the preferred attribute of each object,

$$A_1 = c^*(N_1), \qquad A_2 = c^*(N_2). \tag{A.4}$$

This provides a low-conflict reference setting in which the requested associations and the model's preferences coincide.

Counter-prior. Both requested associations differ from the corresponding object preferences,

$$A_1 \neq c^*(N_1), \qquad A_2 \neq c^*(N_2), \qquad A_1 \neq A_2, \tag{A.5}$$

while exchanging the two requested attributes does not yield a jointly preferred assignment,

$$\neg\left[A_1 = c^*(N_2) \wedge A_2 = c^*(N_1)\right]. \tag{A.6}$$

This condition measures the difficulty introduced by atypical object-attribute combinations without constructing a coherent competing assignment that restores both object preferences.

Reciprocal competitor. Each object receives the other object's preferred attribute,

$$A_1 = c^*(N_2), \qquad A_2 = c^*(N_1), \tag{A.7}$$

with $c^*(N_1) \neq c^*(N_2)$. The requested assignment is therefore

$$\mathcal{B}_{\text{req}} = \{c^*(N_2) \leftrightarrow N_1,\ c^*(N_1) \leftrightarrow N_2\}, \tag{A.8}$$

whereas the complete crossed assignment becomes

$$\mathcal{B}_{\text{cross}} = \{c^*(N_1) \leftrightarrow N_1,\ c^*(N_2) \leftrightarrow N_2\}. \tag{A.9}$$

The requested and crossed outcomes therefore contain exactly the same constituent concepts, while the crossed outcome restores both model-preferred associations. This provides the strongest controlled competition between two complete assignments.

Homogeneous. Both objects are requested with the same attribute,

$$A_1 = A_2 = c_h. \tag{A.10}$$

In this case, exchanging the attributes does not change the requested composition and no distinct crossed assignment exists. The condition therefore controls for the difficulty of jointly maintaining two attributed objects without competition between two alternative assignments.

The purpose of these regimes is diagnostic rather than simply to increase benchmark difficulty. They selectively vary the type of competition faced by the requested correspondence while retaining the same minimal two-object structure.

Benchmark composition. The Controlled Benchmark contains 52 prompt formulations corresponding to 26 semantic configurations evaluated in both constituent orders. Each prompt is generated with 100 seeds, yielding 5,200 generations per backbone. The benchmark comprises the four core compositional regimes above together with additional controlled partner configurations that vary the strength of the competing association. Beyond the four core regimes, we include 10 additional semantic configurations that vary the degree and form of competing attribute pressure introduced by the partner object. These controls use the same two-object template and single-object executability criterion and are evaluated in both constituent orders, yielding 20 additional prompt formulations. They are included in the aggregate Controlled Benchmark results.

A.5 Matched Initial Noise

Prior work has shown that the initial noise realization can strongly influence semantic faithfulness in text-to-image generation, motivating methods that identify or optimize favorable initial latents (Guo et al., 2024). We therefore control explicitly for initial noise when comparing our compositional conditions and ask whether compositional success or failure is primarily determined by the initial noise, or whether the requested attribute-object assignment can change the outcome even when the initialization is held fixed. Each compositional prompt is evaluated over 100 seeds. For matched conditions of a given object pair, we reuse the same set of 100 seeds. Thus, for a fixed seed s, the initial latent is identical while the requested attribute-object assignment is varied across conditions.

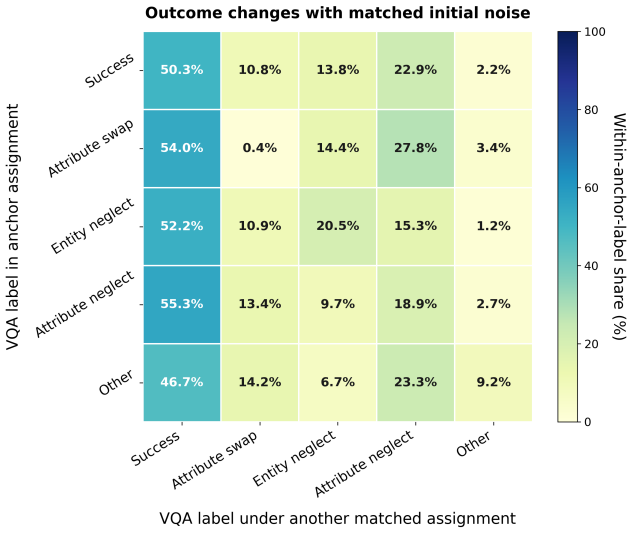

Figure A.5: **Compositional outcomes under matched initial noise.** Each row corresponds to the VQA outcome obtained for a seed under one compositional assignment, while columns show the outcome of the same seed under another matched assignment. The initial latent is therefore held fixed while the requested attribute-object correspondence changes. The substantial off-diagonal mass shows that the same initial noise can lead to different outcomes.

As shown in Fig. A.5 for SD3.5, the same initial noise can lead to different compositional outcomes under different matched assignments: seeds that produce a successful generation in one condition can yield a failure in another, and vice versa. This indicates that initial noise influences the generation trajectory but does not by itself determine compositional success or failure.

A.6 VQA Evaluation and Failure Taxonomy

We use Qwen3-8B as the image evaluator in a visual-question-answering (VQA) setting. The evaluation has two complementary components. First, following the attribute-binding evaluation of T2I-CompBench (Huang et al., 2023), we score the two requested object-attribute pairs independently and combine them using the multiplicative binding score from Eq. (5). Second, we perform a structured diagnostic evaluation to determine how an unsuccessful generation fails.

Structured evaluation. For every generated image, we evaluate the following properties in sequence:

1. **Entity presence:** whether N_1 and N_2 are present in the image.
2. **Identity recognizability:** whether each present entity can be recognized as the requested object independently of its realized attribute.
3. **Attribute scorability:** whether the relevant attribute can be reliably assessed for each recognizable object.
4. **Observed attribute:** the realized attribute associated with each recognizable and scorable object.
5. **Assignment diagnosis:** whether the observed associations match $\mathcal{B}_{\mathrm{req}}$, the complete crossed assignment $\mathcal{B}_{\mathrm{cross}}$, or neither.

Separating object identity from attribute evaluation is important. An object is therefore not considered absent merely because it appears with an unexpected attribute.

Failure categories. Each image is assigned one mutually exclusive diagnostic outcome. The taxonomy is illustrated in Fig. 2.

- **Success:** both requested objects are present and the realized attributes match $\mathcal{B}_{\mathrm{req}}$.
- **Entity neglect:** at least one requested object is absent or cannot be reliably recognized as the requested entity.
- **Attribute swap:** both requested entities are present and recognizable, but their realized attributes match the complete crossed assignment $\mathcal{B}_{\mathrm{cross}}$. This category is defined only when a distinct crossed assignment exists.
- **Attribute neglect:** the requested entities are retained, but at least one requested attribute is not successfully realized, without forming the complete crossed assignment.
- **Other:** failures that do not admit a reliable assignment-level diagnosis. This includes ambiguous, occluded, cropped, or fused generations, as well as scorable outcomes that satisfy neither the requested nor the complete crossed assignment.

We apply the categories hierarchically such that every image receives exactly one diagnostic label. In particular, entity-level failures are separated from attribute-level errors, and a sample is classified as an attribute swap only when the complete crossed assignment is realized. The diagnostic labels are assigned only after generation. They are used to characterize benchmark outcomes and to group trajectories for the mechanistic analyses in Sec. 5; they are not provided to the generative model or used to modify its internal states.

B Additional Analysis of Compositional Assignment Dynamics

This section provides additional details and full block-wise analyses for the trajectory-level measurements introduced in Sec. 5. We first describe how trajectories are grouped by their final generation outcome. We then report the assignment geometry across transformer depth and denoising time, followed by the complete analysis of depth-dependent text updates.

B.1 Trajectory Grouping and Analysis Protocol

For every prompt, we generate 100 samples using different initial noise seeds, as described in Sec. 4. Each generated image is assigned one of the diagnostic outcome labels defined in Sec. A.6. These labels are determined exclusively from the final generated image and are not used during generation

or in the computation of the internal representations. For the mechanistic comparison, we condition on the selected failure category and retain prompts containing at least one seed with that failure. Successful trajectories are drawn from successful seeds of the same prompts when available.

For each retained seed, we record the relevant internal quantities at every considered transformer block l and denoising step t. Representation-level measurements are computed independently for each seed before aggregating trajectories according to their final diagnostic outcome.

B.2 Assignment Geometry Across Blocks and Denoising Time

The assignment geometry introduced in Sec. 5.1 provides a direct representation-level view of how each noun is organized relative to the two attributes. Recall that the attribute direction $\mathbf{d}_A^{l,t}$ is oriented from A_1 toward A_2, and that the corresponding noun projections are

$$s_i^{l,t} = \left\langle \mathbf{x}_{N_i}^{l,t} - \frac{1}{2}\left(\mathbf{x}_{A_1}^{l,t} + \mathbf{x}_{A_2}^{l,t}\right), \mathbf{d}_A^{l,t} \right\rangle, \qquad i \in \{1,2\}. \tag{B.1}$$

Under the requested correspondence, N_1 should lie toward A_1 and N_2 toward A_2, corresponding to

$$s_1^{l,t} < 0 < s_2^{l,t}. \tag{B.2}$$

Conversely, a simultaneous reversal of these positions is consistent with the crossed correspondence. We retain the two noun projections separately rather than collapsing them into a single scalar margin. This is important because a change in correspondence need not occur symmetrically: one noun may move substantially toward the competing attribute while the other remains relatively stable.

We report the full block-wise assignment geometry separately for the three failure modes. For SD3.5, results are shown for both the T5 and CLIP text representations: attribute swaps in Figs. B.1 and B.2, entity neglect in Figs. B.3 and B.4, and attribute neglect in Figs. B.5 and B.6. The corresponding FLUX.1-dev results using the T5 representations are shown for attribute swaps, entity neglect, and attribute neglect in Figs. B.7 to B.9, respectively.

Observed dynamics. For SD3.5, attribute swaps exhibit the clearest depth-dependent reorganization. In both the T5 and CLIP representations, the requested noun ordering is still present at the beginning of the trajectory in intermediate blocks, but progressively reverses during denoising. This transition becomes particularly visible around blocks 7–14, while at deeper blocks the crossed organization dominates a larger portion of the trajectory. The qualitatively similar behavior across the two text encoders indicates that the reorganization is not specific to a single conditioning stream.

The neglect modes show a different pattern. For entity and attribute neglect, successful and failed trajectories remain considerably closer than for attribute swaps and generally do not form a coherent crossed assignment. Differences are instead more apparent toward later denoising steps, where the noun-attribute separation becomes weaker or more variable. Thus, the assignment-axis geometry most clearly exposes a structured reassignment for swaps, while neglect is associated with a less pronounced loss of relational organization.

For FLUX.1-dev, the swap transition appears later in network depth. The requested ordering is largely maintained through the earlier blocks, begins to destabilize around blocks 17–18, and becomes increasingly consistent with the crossed assignment in subsequent blocks. Entity-neglect trajectories, in contrast, remain close to successful trajectories across most blocks. Attribute neglect likewise shows limited separation initially, but develops more localized differences at later depth, particularly around blocks 40 and 48. These results suggest that different failure modes need not share a single depth at which their internal trajectories diverge.

B.3 Depth-Dependent Text Updates

The absolute hidden representations contain substantial information shared by all prompt constituents as they are repeatedly transformed through the network. To focus on constituent-specific changes, the analysis in Sec. 5.1 first measures the displacement of each constituent relative to its representation at the input of the transformer,

$$\mathbf{r}_X^{l,t} = \mathbf{x}_X^{l,t} - \mathbf{x}_X^{0,t}, \tag{B.3}$$

and then removes the displacement component shared across the four constituents,

$$\widetilde{\mathbf{r}}_X^{l,t} = \mathbf{r}_X^{l,t} - \frac{1}{|\mathcal{C}|} \sum_{Y \in \mathcal{C}} \mathbf{r}_Y^{l,t}, \qquad \mathcal{C} = \{A_1, N_1, A_2, N_2\}. \tag{B.4}$$

This centering isolates differences in how the individual constituents are updated, rather than changes shared by the complete prompt representation.

For each attribute-noun pair, we measure the cosine similarity between their centered updates,

$$\gamma_{ij}^{l,t} = \cos\left(\widetilde{\mathbf{r}}_{A_i}^{l,t}, \widetilde{\mathbf{r}}_{N_j}^{l,t}\right). \tag{B.5}$$

The text-update margin

$$M_{l,t} = \frac{1}{2}\left[\left(\gamma_{11}^{l,t} - \gamma_{21}^{l,t}\right) + \left(\gamma_{22}^{l,t} - \gamma_{12}^{l,t}\right)\right] \tag{B.6}$$

compares the similarity of the requested attribute-noun updates against the corresponding crossed alternatives. Positive values indicate relatively stronger requested-pair structure, negative values indicate relatively stronger crossed-pair structure, and values near zero indicate that neither pairing is strongly differentiated.

We report the full block-wise text-update margin separately for the three failure modes. For SD3.5, results are shown for both the T5 and CLIP text representations: attribute swaps in Figs. B.10 and B.11, entity neglect in Figs. B.12 and B.13, and attribute neglect in Figs. B.14 and B.15. The corresponding FLUX.1-dev results using the T5 representations are shown for attribute swaps, entity neglect, and attribute neglect in Figs. B.16 to B.18, respectively.

Observed dynamics. Across failure modes, failed trajectories generally exhibit a lower text-update margin than successful trajectories, indicating weaker relative support for the requested pairing. For SD3.5, this separation is already visible around block 7 for both T5 and CLIP and is strongest for attribute swaps. At later blocks, the gap increases and failed swap trajectories can approach or enter the crossed-pair regime. The same tendency is present but weaker for entity and attribute neglect, where successful and failed trajectories overlap more strongly.

For FLUX.1-dev, the earliest clear separation appears around block 17, particularly for attribute swaps. Entity neglect shows little outcome-dependent separation, while attribute neglect exhibits a weaker and more localized difference at later blocks. These observations motivate the early detector locations used in Sec. C.1: block 7 for SD3.5 and block 17 for FLUX.1-dev.

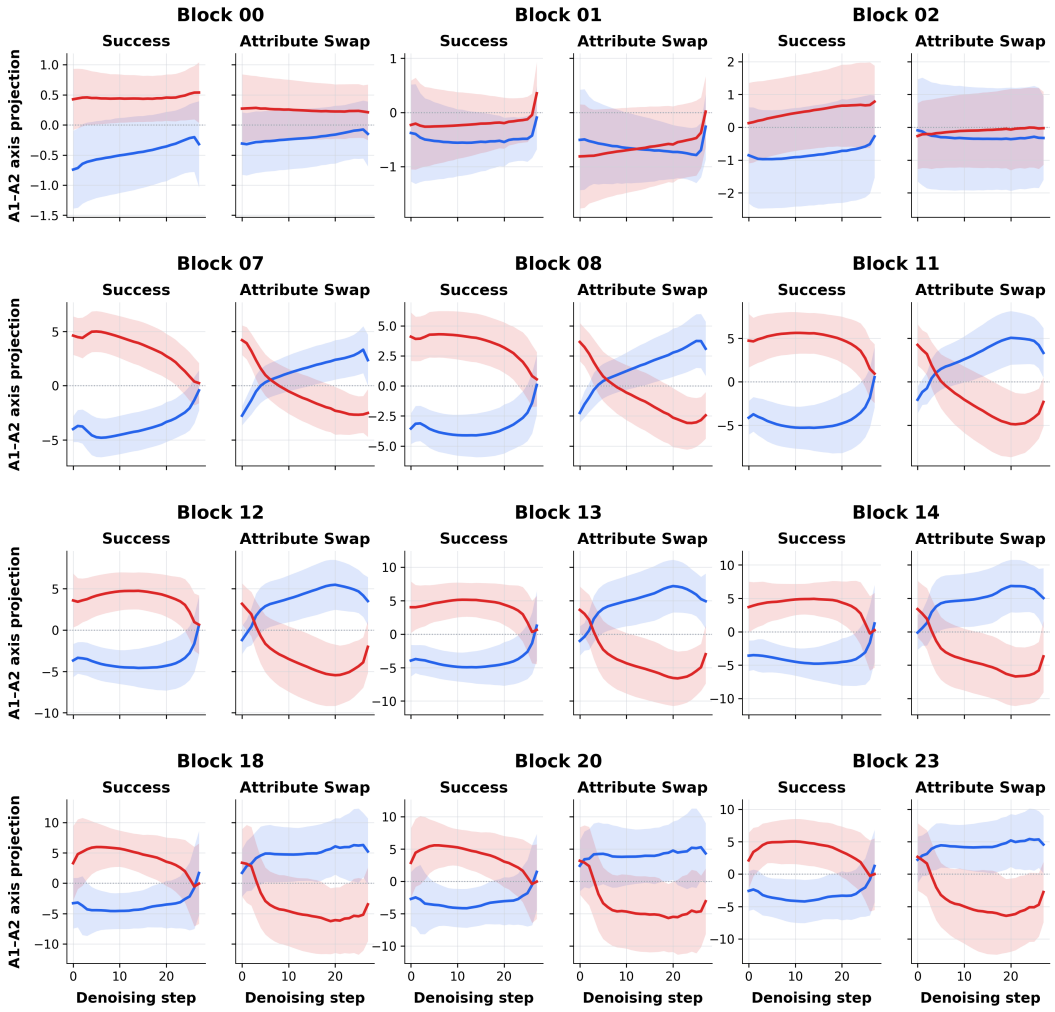

Figure B.1: **Block-wise assignment geometry for attribute swaps in SD3.5 (T5 encoder).** Noun projections onto the evolving A_1–A_2 attribute axis are shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories. Shaded regions indicate variability across generations.

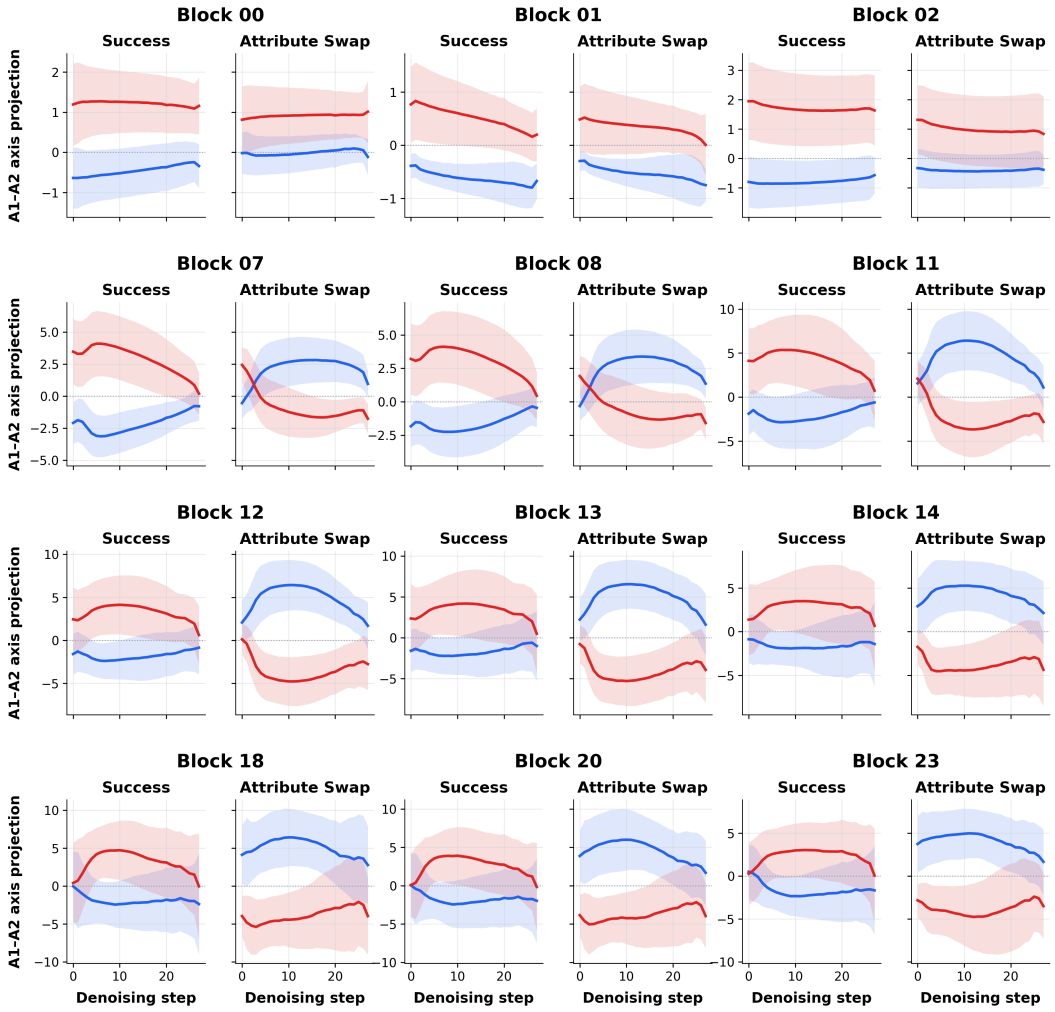

Figure B.2: **Block-wise assignment geometry for attribute swaps in SD3.5 (CLIP encoder).** Noun projections computed from the CLIP text representations are shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories. Shaded regions indicate variability across generations.

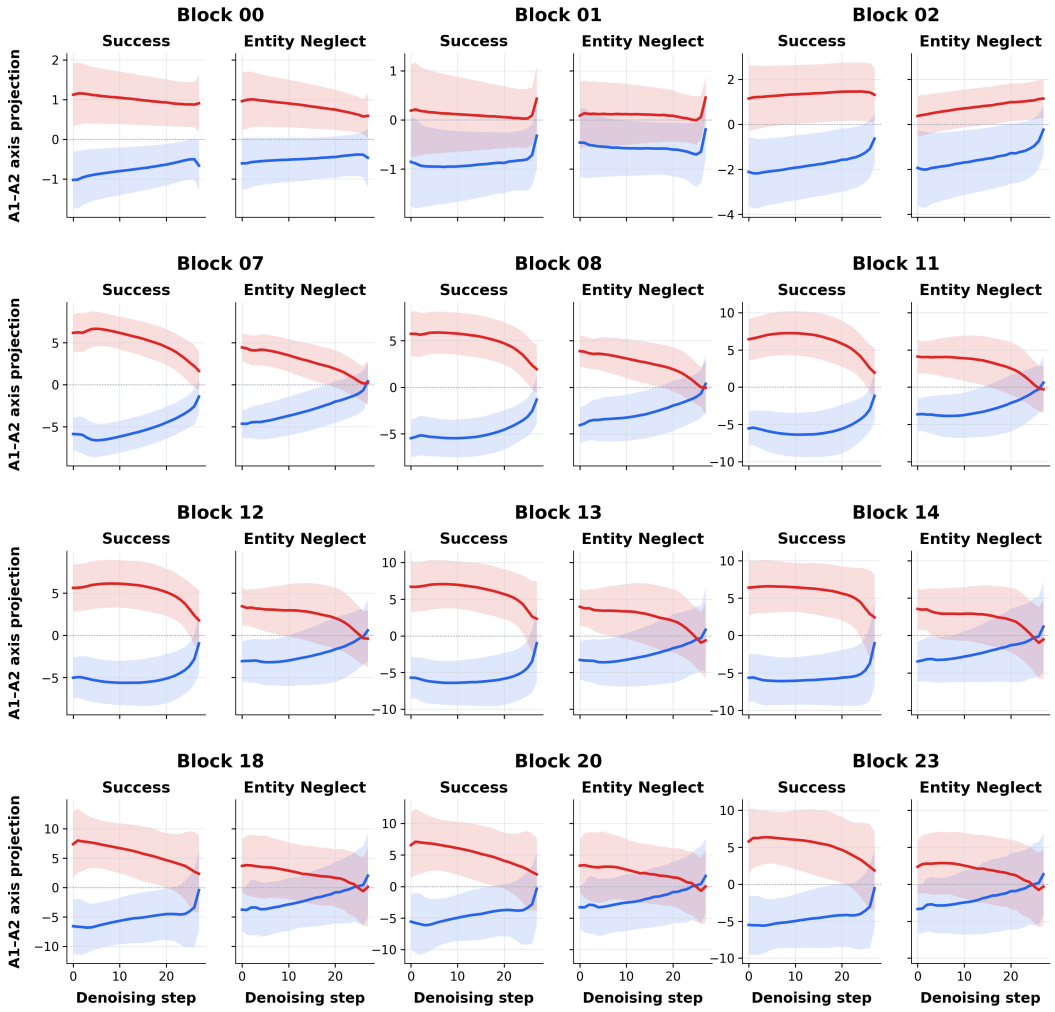

Figure B.3: **Block-wise assignment geometry for entity neglect in SD3.5 (T5 encoder).** Noun projections onto the evolving attribute axis are shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories. Shaded regions indicate variability across generations.

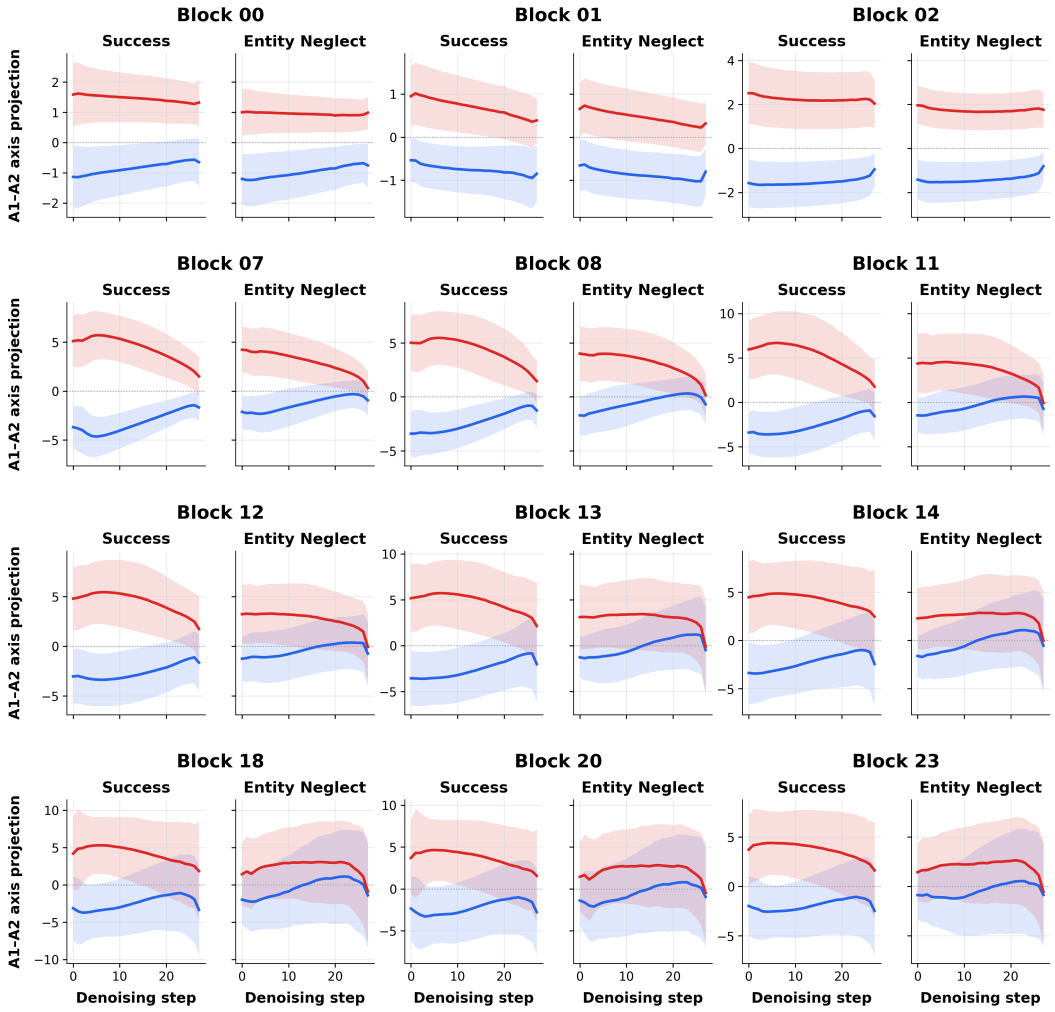

Figure B.4: **Block-wise assignment geometry for entity neglect in SD3.5 (CLIP encoder).** Noun projections computed from the CLIP text representations are shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories. Shaded regions indicate variability across generations.

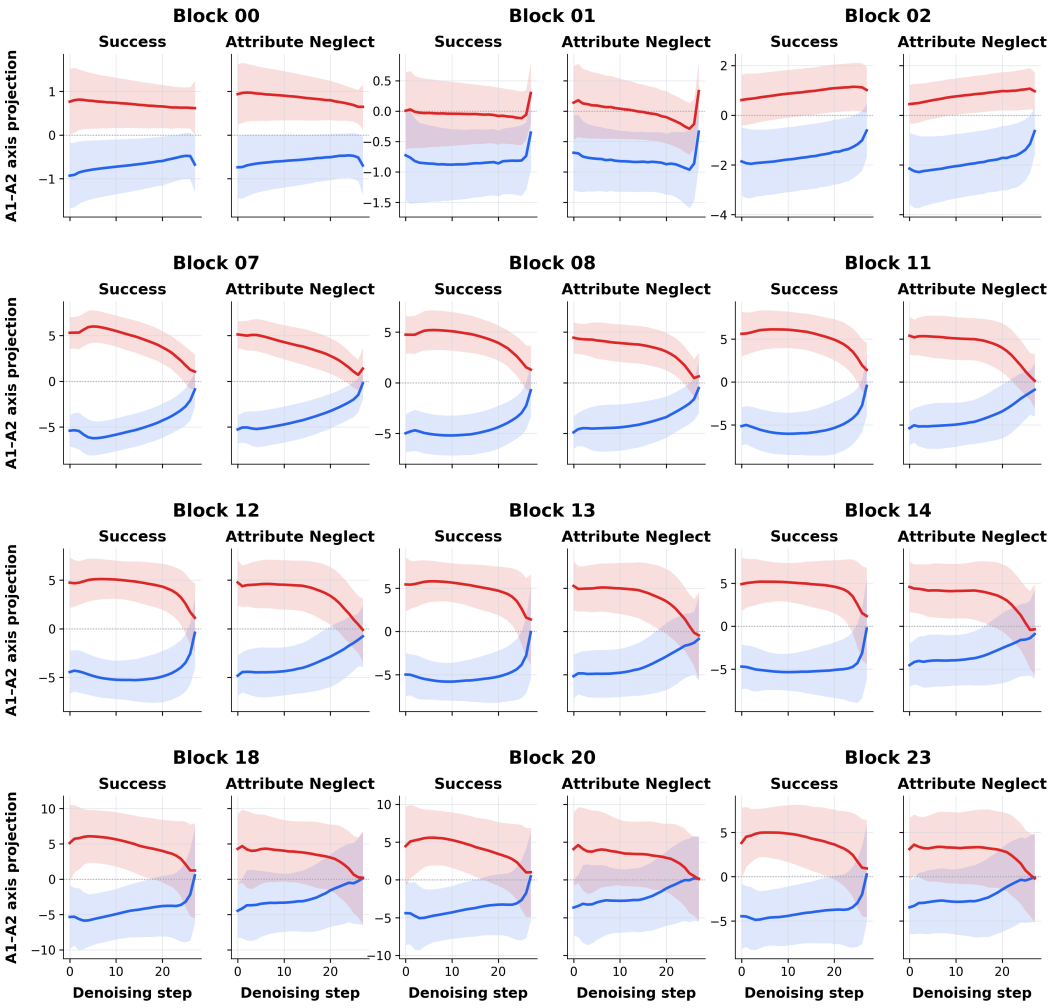

Figure B.5: **Block-wise assignment geometry for attribute neglect in SD3.5 (T5 encoder).** Noun projections onto the evolving attribute axis are shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories. Shaded regions indicate variability across generations.

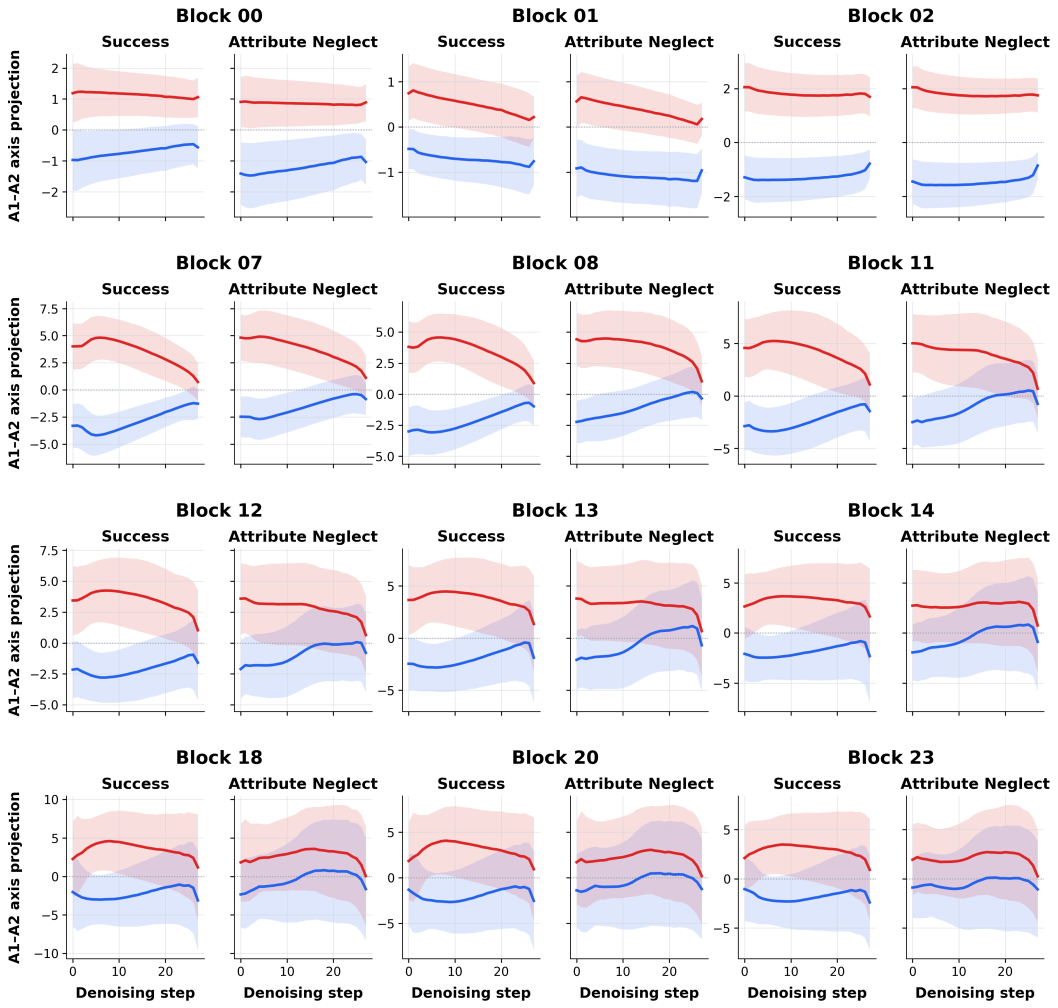

Figure B.6: **Block-wise assignment geometry for attribute neglect in SD3.5 (CLIP encoder).** Noun projections computed from the CLIP text representations are shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories. Shaded regions indicate variability across generations.

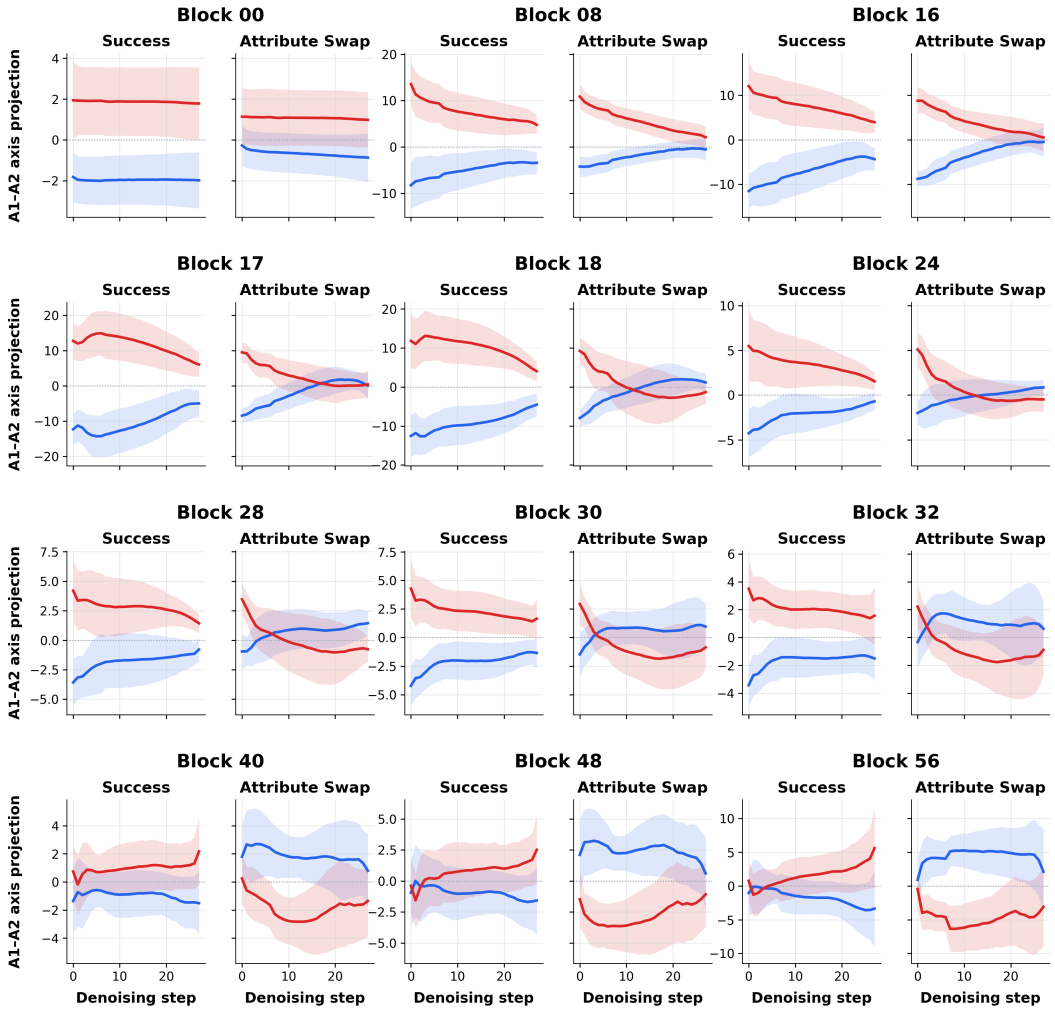

Figure B.7: **Block-wise assignment geometry for attribute swaps in FLUX.1-dev.** Noun projections from the T5 text representations are shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories. Shaded regions indicate variability across generations.

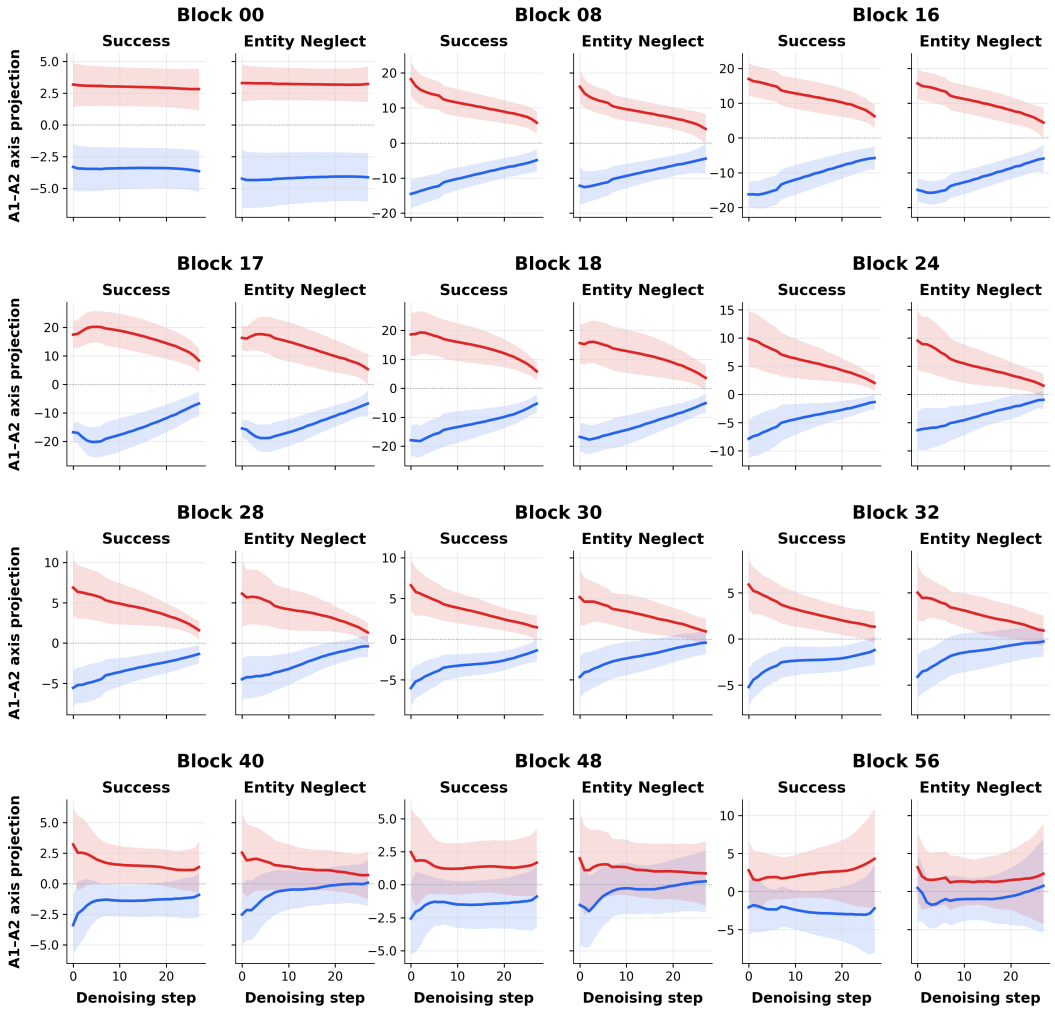

Figure B.8: **Block-wise assignment geometry for entity neglect in FLUX.1-dev.** Noun projections from the T5 text representations are shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories. Shaded regions indicate variability across generations.

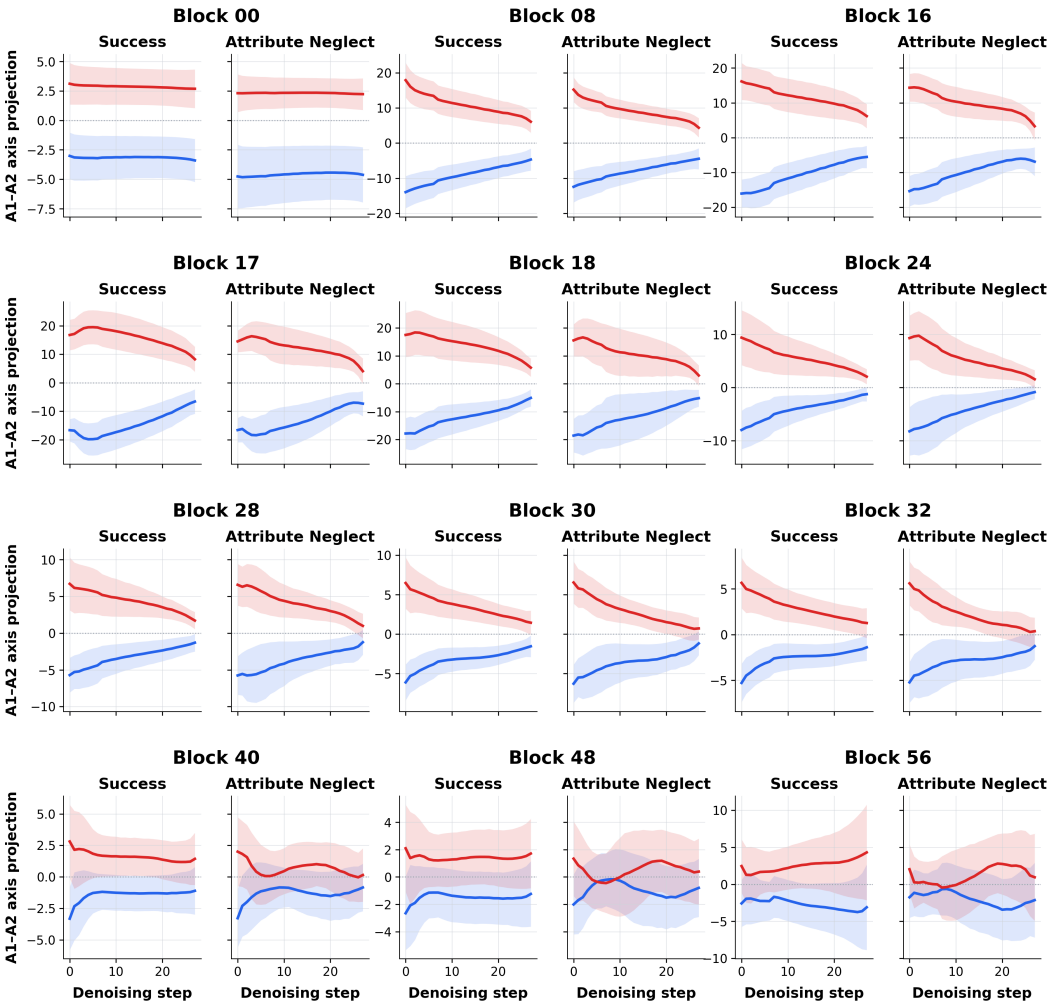

Figure B.9: **Block-wise assignment geometry for attribute neglect in FLUX.1-dev.** Noun projections from the T5 text representations are shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories. Shaded regions indicate variability across generations.

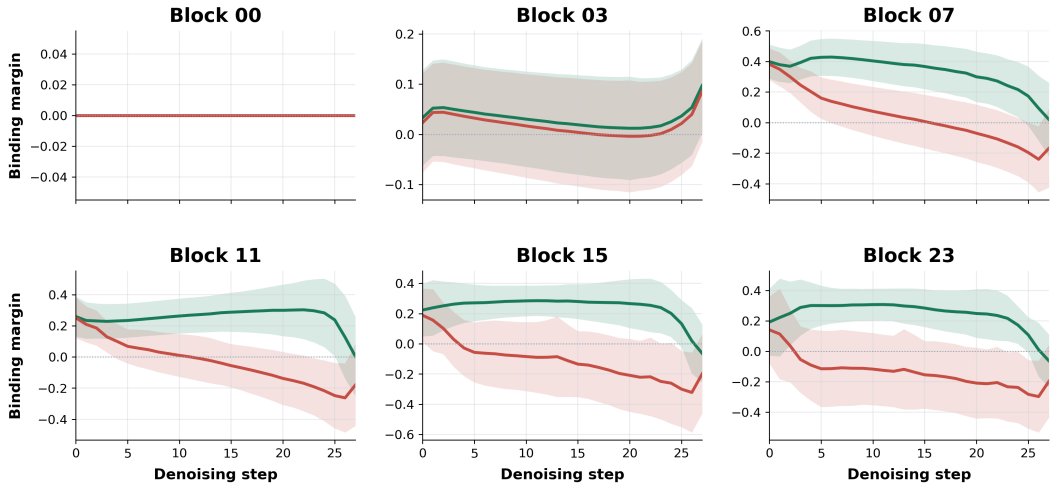

Figure B.10: **Depth-dependent text-update margin for attribute swaps in SD3.5 (T5 encoder).** The text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories. Positive values indicate relatively stronger requested-pair structure, while negative values indicate relatively stronger crossed-pair structure.

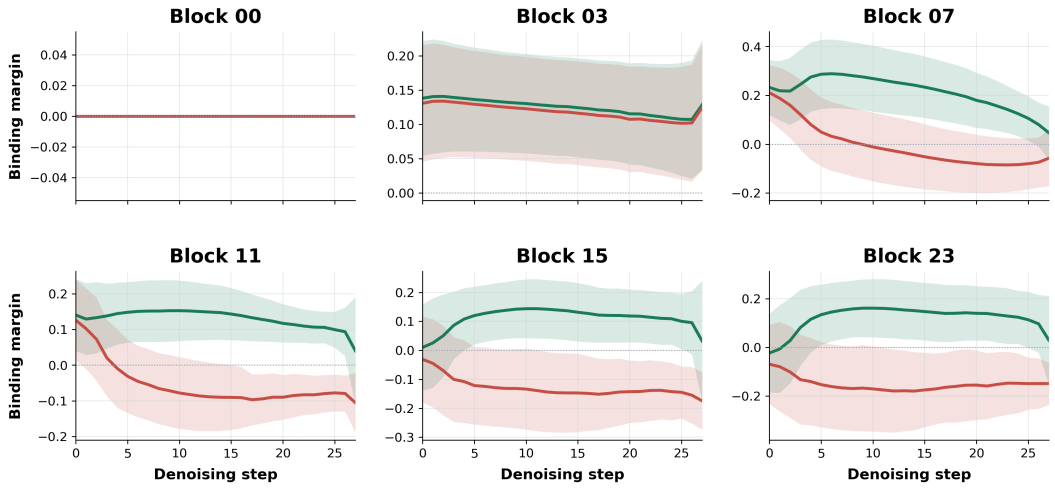

Figure B.11: **Depth-dependent text-update margin for attribute swaps in SD3.5 (CLIP encoder).** The text-update margin $M_{l,t}$ computed from the CLIP representations is shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories.

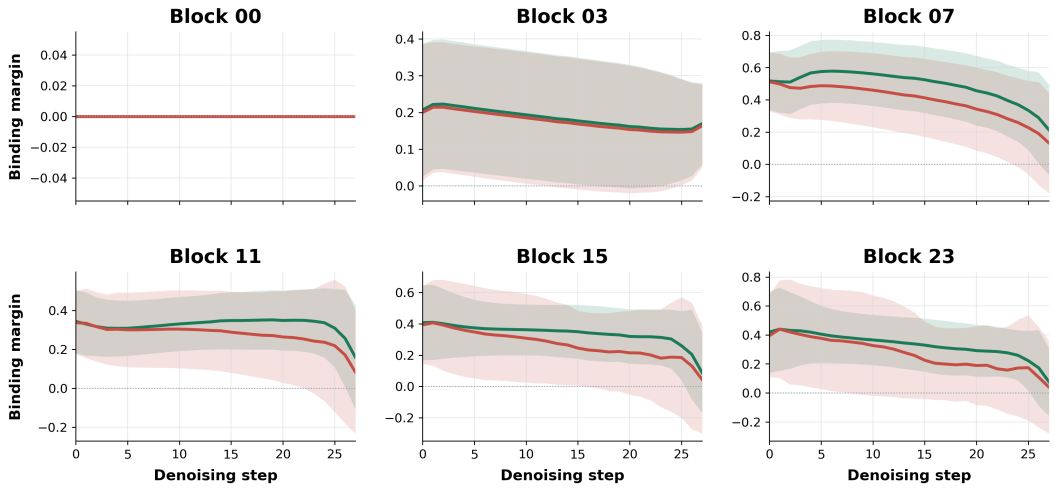

Figure B.12: **Depth-dependent text-update margin for entity neglect in SD3.5 (T5 encoder).** The text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories. Values near zero indicate weak differentiation between the requested and crossed pairings.

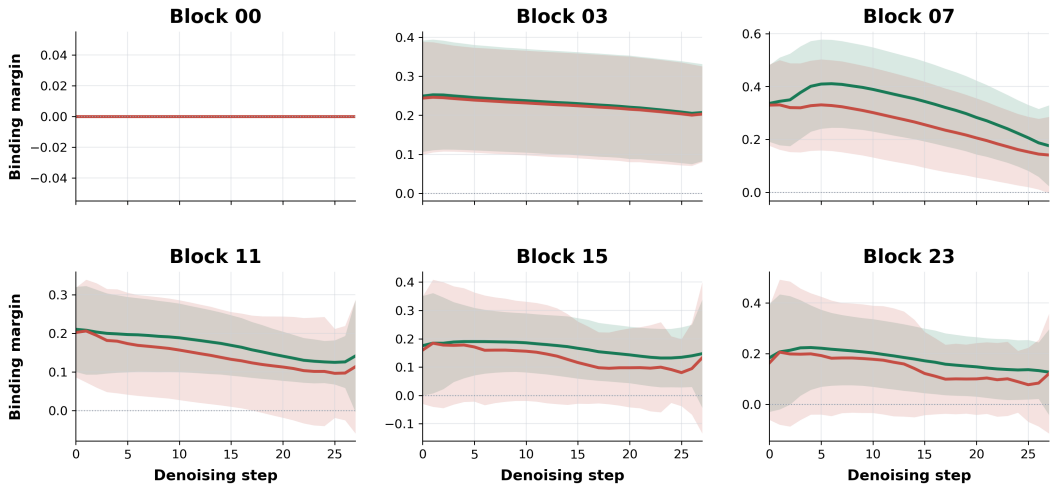

Figure B.13: **Depth-dependent text-update margin for entity neglect in SD3.5 (CLIP encoder).** The text-update margin $M_{l,t}$ computed from the CLIP representations is shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories.

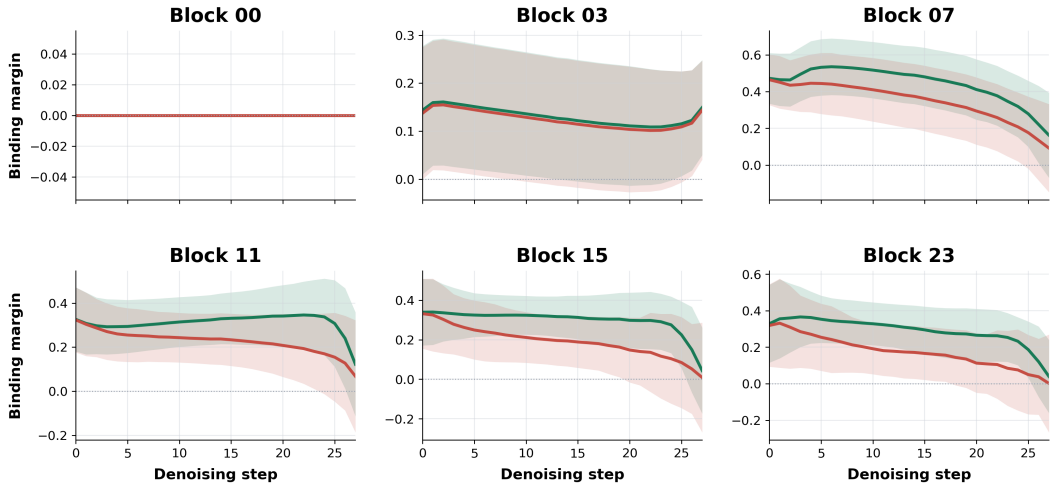

Figure B.14: **Depth-dependent text-update margin for attribute neglect in SD3.5 (T5 encoder).** The text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories. Values near zero indicate weak differentiation between the requested and crossed pairings.

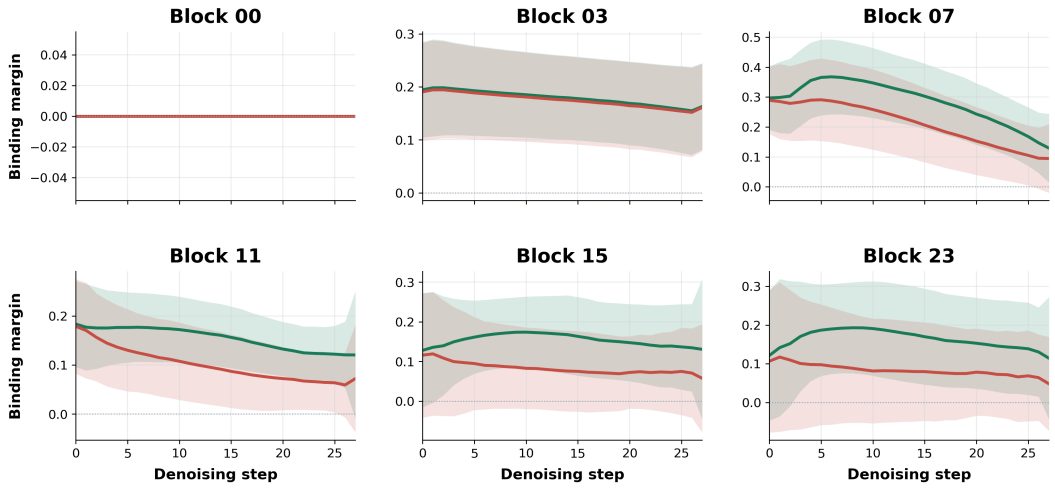

Figure B.15: **Depth-dependent text-update margin for attribute neglect in SD3.5 (CLIP encoder).** The text-update margin $M_{l,t}$ computed from the CLIP representations is shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories.

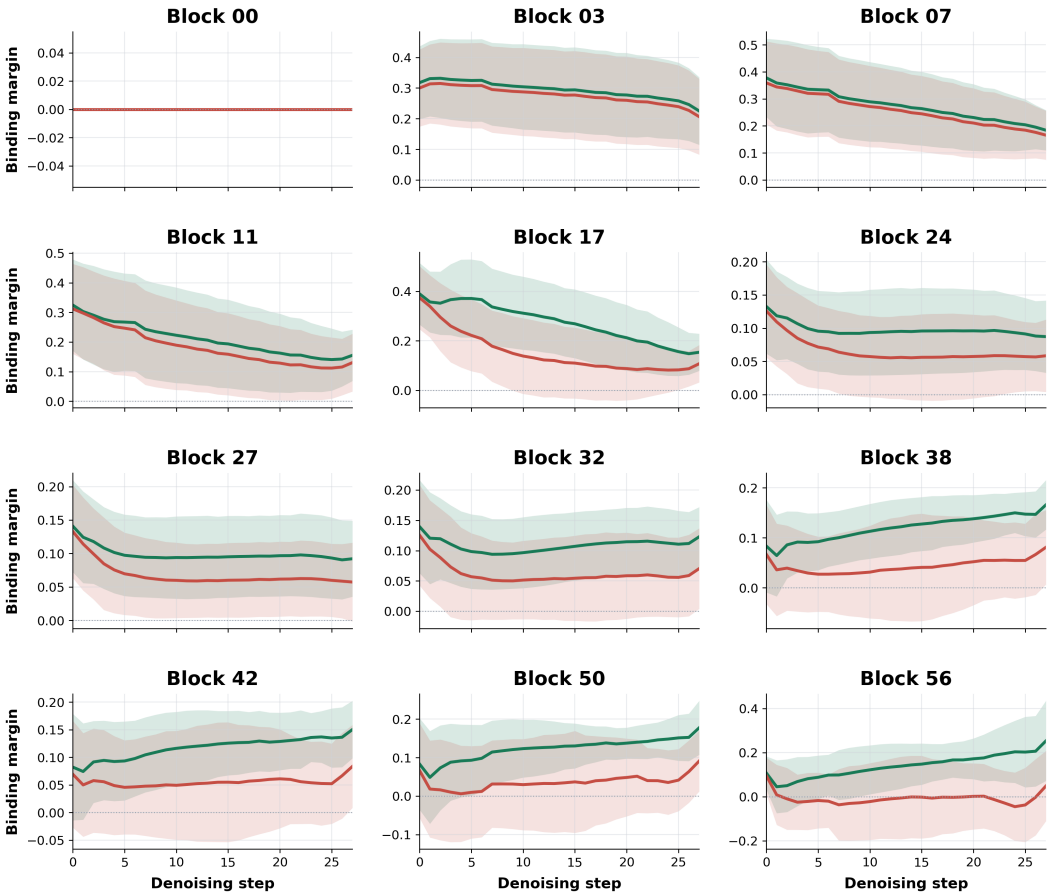

Figure B.16: **Depth-dependent text-update margin for attribute swaps in FLUX.1-dev.** The T5-based text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and attribute-swap trajectories. Positive and negative values indicate relative support for the requested and crossed pairings, respectively.

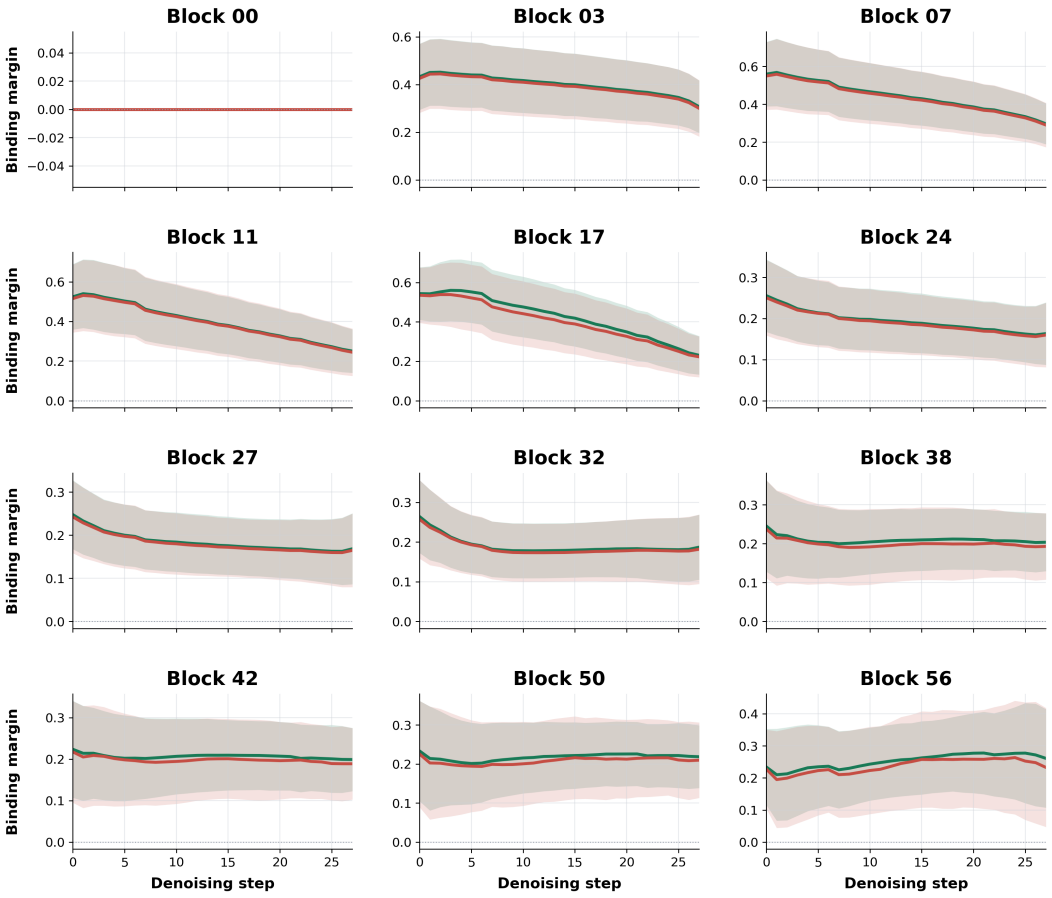

Figure B.17: **Depth-dependent text-update margin for entity neglect in FLUX.1-dev.** The T5-based text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and entity-neglect trajectories.

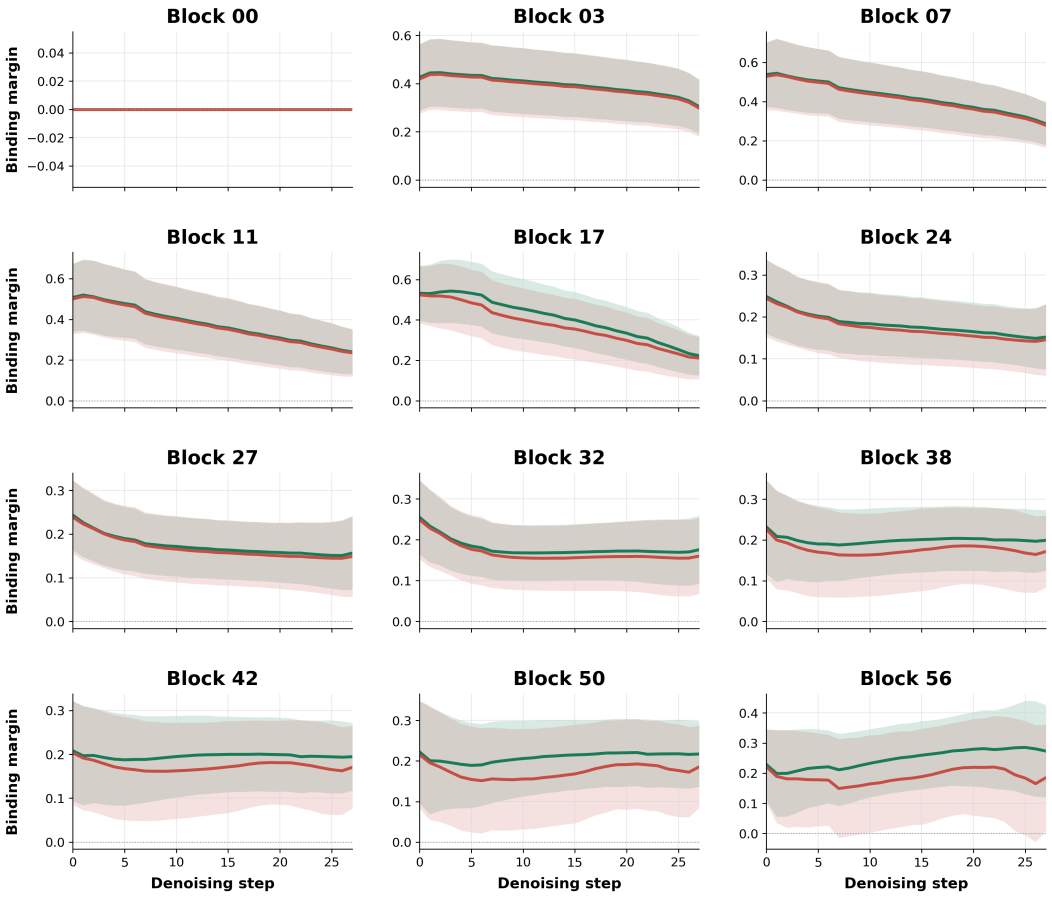

Figure B.18: **Depth-dependent text-update margin for attribute neglect in FLUX.1-dev.** The T5-based text-update margin $M_{l,t}$ is shown across denoising time for selected transformer blocks, comparing successful and attribute-neglect trajectories.

C Intervention Details and Additional Results

This section provides the implementation details and additional results for the intervention introduced in Sec. 6. We report the exact construction of the early failure detector, the model-specific intervention configurations, and additional results omitted from the main paper.

C.1 Early Failure Detection

The failure detector operates on the text-update margin $M_{l,t}$ defined in Eq. (10). For each backbone, we evaluate $M_{l,t}$ at a small set of early block-timestep locations and aggregate these measurements into the scalar score R used for intervention selection in Sec. 6.1. All block and denoising-step indices below are zero-based.

For SD3.5, let $\widehat{F}_e$ be the empirical cumulative distribution function of $-M^e_{7,5}$ among the baseline successes for text scope $e \in \{\mathrm{T5}, \mathrm{CLIP}\}$. We use

$$R_{\mathrm{SD3.5}} = \frac{1}{2}\left[\widehat{F}_{\mathrm{T5}}\big(-M^{\mathrm{T5}}_{7,5}\big) + \widehat{F}_{\mathrm{CLIP}}\big(-M^{\mathrm{CLIP}}_{7,5}\big)\right]. \tag{C.1}$$

Here, each margin is computed from the block-7 text input relative to the block-0 text input within the same step-5 forward pass. The detector requests intervention when $R_{\mathrm{SD3.5}} > \tau_{\mathrm{SD3.5}}$, where $\tau_{\mathrm{SD3.5}} = 0.7$.

For FLUX.1-dev, we use the mean T5 margin over the first five replayable denoising steps,

$$R_{\mathrm{FLUX}} = \frac{1}{5}\sum_{t=1}^{5} M^{\mathrm{T5}}_{17,t}. \tag{C.2}$$

Each term is computed from the block-17 text input relative to block 0 in the same forward pass. Because a smaller margin indicates greater risk, the FLUX detector requests intervention when $R_{\mathrm{FLUX}} < \tau_{\mathrm{FLUX}}$, where $\tau_{\mathrm{FLUX}} = 0.53$. For either backbone, a missing or non-finite required observation conservatively triggers intervention.

Detector calibration and audit. The detector design is based on the baseline trajectory cohort. The selected block-timestep statistics, temporal aggregation, and intervention-rate thresholds are determined from these baseline trajectories without fitting an auxiliary classifier. We additionally perform a prompt-held-out audit in which detector statistics are evaluated on prompts excluded from the corresponding reference fold. This audit measures whether the selected signal retains discriminative value across prompts and estimates the resulting success-trajectory flag rate; it is not the evaluation protocol used for the intervention results reported in Sec. 6.3.

For the reported intervention experiments, the deployed detector is subsequently instantiated using the complete baseline-success cohort. For SD3.5, the empirical CDFs of $-M^e_{7,5}$ are computed from all baseline successes for $e \in \{\mathrm{T5}, \mathrm{CLIP}\}$, and intervention is requested when $R_{\mathrm{SD3.5}} > 0.70$. For FLUX.1-dev, we use the mean block-17 T5 margin over the first five denoising steps and intervene below the 26th percentile of this statistic among baseline successes. Thus, the reported intervention results constitute a same-cohort evaluation of whether the diagnosed assignment geometry can be used for steering, rather than a held-out evaluation of detector generalization. Still the same parameters are used for evaluationg on T2I-CompBench++.

C.2 Model-Specific Intervention Configuration

All intervention experiments use the same model and sampling configuration as the corresponding baseline generations described in App. Sec. A; only the detector and steering settings are changed. The intervention formulation is shared across backbones, while the internal states and optimization schedule are selected separately for SD3.5 and FLUX.1-dev. The complete configurations are summarized in Tab. C.1.

At the anchor step, only blocks satisfying $s_1^{l,t_a} < 0 < s_2^{l,t_a}$ are retained. A rejected block contributes no loss; if no block supplies a valid anchor, the latent update is skipped and ordinary denoising continues. During guided denoising, the active loss terms from all retained blocks and text scopes

	SD3.5	FLUX.1-dev
Scout steps	1–5	1–5
Detector blocks	0, 7 (T5 and CLIP)	0, 17 (T5)
Detector timesteps	5	1–5
Threshold τ	0.70 ($R > \tau$)	0.53 ($R < \tau$)
Anchor step t_a	0	0
Margin scale η	0.8	0.2
Guided steps	1–10	1–5
Steering strength α	3	5

Table C.1: **Model-specific intervention configuration.** Detector, anchor, and steering settings used for SD3.5 and FLUX.1-dev.

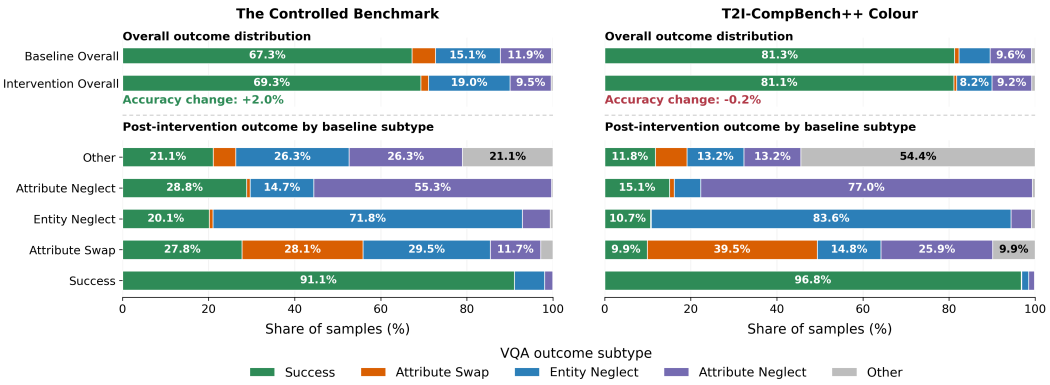

Figure C.1: **Outcome transitions under selective self-anchored intervention on FLUX.1-dev.** Top: overall VQA outcome distributions before and after selective intervention on the Controlled Benchmark and T2I-CompBench++ Colour. Bottom: post-intervention outcomes conditioned on the baseline VQA category. Selective steering preserves most baseline successes, although its overall correction effect is weaker than for SD3.5.

are summed before differentiation. We apply one normalized-gradient latent update per guided denoising step, $z_t \leftarrow z_t - \alpha \nabla_{z_t} L / \|\nabla_{z_t} L\|_2$, whenever the aggregate loss and gradient are finite and nonzero.

C.3 Additional FLUX.1-dev Results

We additionally evaluate the intervention on FLUX.1-dev using the configuration in Tab. C.1, as represented in Fig. C.1. On the Controlled Benchmark, success changes from 67.3% to 69.3%. On T2I-CompBench++ Colour, success changes from 81.3% to 81.1%, a loss of -0.2%.

The intervention therefore transfers less strongly to FLUX.1-dev than to SD3.5. While the assignment geometry remains diagnostically informative, its presence does not imply that the corresponding state is equally controllable through the image latent across architectures.

C.4 Effect of Selective Intervention

The main method applies steering only to trajectories identified as at risk by the early failure detector. To isolate the role of this filtering step, we additionally evaluate an *always-on* variant in which the same self-anchored steering is applied to every trajectory, independently of its failure-risk score. The complete outcome transitions are shown in Figs. C.2 and C.3 for SD3.5 and FLUX.1-dev, respectively.

For SD3.5, always-on steering remains highly effective on our Controlled Benchmark, increasing the success rate from 64.3% to 84.8% ($+20.4\%$). In this challenging setting, the intervention corrects a large fraction of every baseline failure subtype: 74.3% of attribute swaps, 72.1% of attribute-

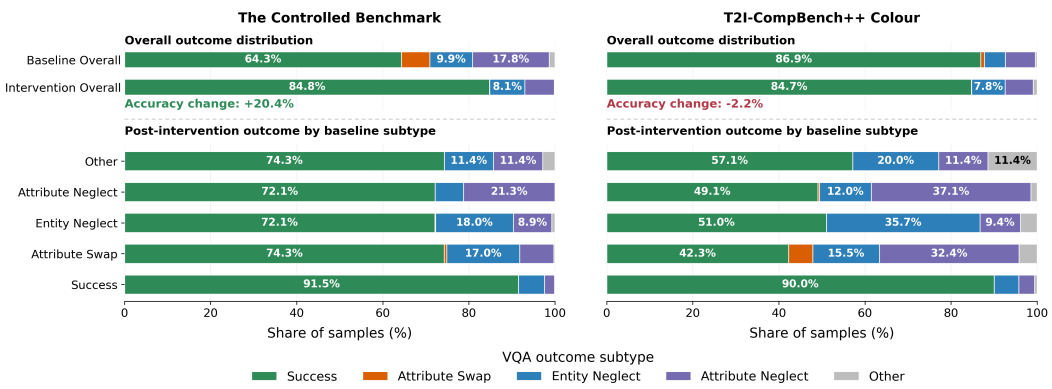

Figure C.2: **Outcome transitions under always-on self-anchored intervention on SD3.5.** The same steering used by the selective method is applied to every trajectory, without failure-based selection. Top: overall outcome distributions before and after intervention. Bottom: post-intervention outcomes conditioned on the baseline VQA category. Always-on steering remains highly effective on the Controlled Benchmark but causes more regressions among already-successful T2I-CompBench++ trajectories.

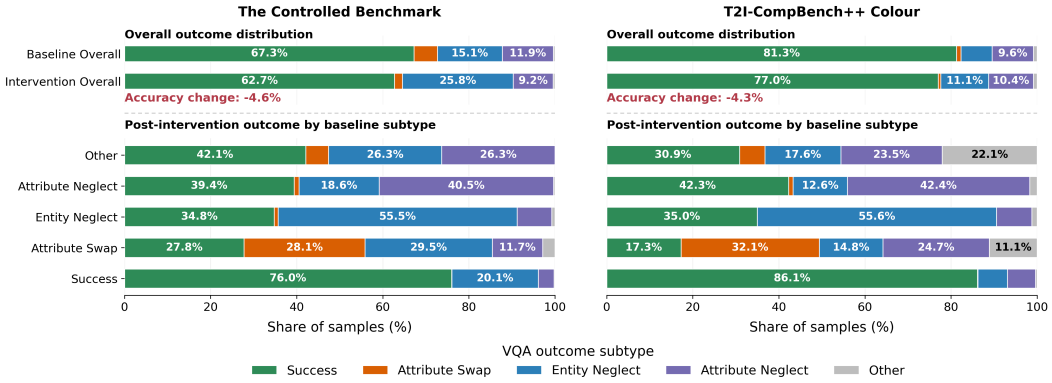

Figure C.3: **Outcome transitions under always-on self-anchored intervention on FLUX.1-dev.** Steering is applied to every trajectory without failure-based selection. Top: overall outcome distributions before and after intervention. Bottom: post-intervention outcomes conditioned on the baseline VQA category. Although a subset of baseline failures is corrected, regressions among initially successful trajectories reduce overall compositional success on both benchmarks.

neglect failures, 72.1% of entity-neglect failures, and 74.3% of samples assigned to OTHER become successful. The effect is different on T2I-CompBench++ Colour, where the SD3.5 baseline is already substantially stronger. Applying steering indiscriminately decreases the overall success rate from 86.9% to 84.7% (-2.2%). Although always-on steering still corrects a substantial fraction of baseline failures, only 90.0% of baseline successes remain successful, compared with 97.9% under selective intervention. Thus, when many trajectories are already correct, unnecessary steering can outweigh the corrections obtained on failing samples. This behavior is even more pronounced for FLUX.1-dev. Without failure-based filtering, success decreases from 67.3% to 62.7% on the Controlled Benchmark and from 81.3% to 77.0% on T2I-CompBench++ Colour. In particular, only 76.0% and 86.1% of baseline-successful trajectories, respectively, remain successful after always-on steering. The intervention therefore remains capable of correcting individual failures, but these gains are outweighed by regressions on trajectories that did not require intervention.

We additionally compare the effect of selective and always-on steering on overall image quality. Interestingly, the loss in compositional success under always-on steering is not accompanied by a corresponding degradation in FID. As shown in Tab. C.2, both intervention variants generally preserve or improve FID relative to the baseline. Always-on steering achieves the lowest FID on the

Benchmark	Model	Baseline	Selective	Always-on
Controlled	SD3.5	165.5	158.9	**156.5**
	FLUX.1-dev	161.9	153.0	**142.9**
T2I-CompBench++ Colour	SD3.5	74.3	73.6	**71.8**
	FLUX.1-dev	75.7	**75.4**	76.2

Table C.2: **Image quality under selective and always-on intervention.** FID (↓) for baseline generation, selective intervention, and always-on intervention without failure-based filtering. All variants are evaluated on the same prompt-seed sets.

Controlled Benchmark for both SD3.5 and FLUX.1-dev, as well as on T2I-CompBench++ Colour for SD3.5. The only exception is FLUX.1-dev on T2I-CompBench++ Colour, where selective steering obtains a slightly lower FID (75.4 versus 76.2). This result further separates visual quality from compositional correctness.

Together, these results separate the effect of the steering objective from the effect of trajectory selection. On the difficult SD3.5 Controlled Benchmark, self-anchored steering is effective even without filtering, providing a 20.4% improvement. Failure-based selection becomes important for avoiding unnecessary changes to already-correct trajectories, particularly on the higher-performing T2I-CompBench++ setting and for FLUX.1-dev. Importantly, Tab. C.2 shows that these compositional regressions are not explained by a general loss of visual quality: always-on steering often achieves equal or better FID despite lower compositional success. We therefore use the selective scout-and-replay variant for the main intervention results.

C.5 Directional Intervention Controls

Direction	α	Success	Swap	Entity Neglect	Attribute Neglect	Other
Crossed-target	1	28.3	51.5	7.7	10.7	1.8
	3	7.0	**83.8**	3.3	5.9	0.0
	5	1.5	**89.3**	2.6	5.1	1.5
Random	1	67.3	9.2	13.6	8.8	1.1
	3	47.8	12.9	26.8	11.4	1.1
	5	40.1	14.3	29.4	15.1	1.1

Table C.3: **Directional intervention controls on SD3.5.** VQA outcome distributions (%) after steering baseline-successful generations toward the crossed assignment or along a random direction. Crossed-target steering predominantly induces attribute swaps, with the effect increasing with steering strength, whereas random-direction steering produces substantially fewer swaps and more diffuse failure modes.

To test whether the intervention acts specifically through the diagnosed assignment geometry rather than through a generic perturbation of the latent trajectory, we perform two controls on baseline-successful SD3.5 generations. In the *crossed-target* control, we reverse the steering objective toward the competing assignment. In the *random-direction* control, we apply a perturbation of the same strength along a random direction.

As shown in Tab. C.3, crossed-target steering predominantly induces complete attribute swaps, with the effect strengthening as the steering scale increases: the swap rate rises from 51.5% at $\alpha = 1$ to 83.8% at $\alpha = 3$ and 89.3% at $\alpha = 5$. In contrast, random-direction steering produces far fewer swaps (9.2%, 12.9%, and 14.3%, respectively) and instead yields a broader mixture of entity and attribute neglect.

These controls show that the intervention effect is strongly direction-specific: steering toward the competing assignment selectively induces the corresponding crossed composition, whereas perturbing the trajectory in a random direction causes substantially more diffuse compositional degradation.

C.6 Additional Qualitative Results

We provide additional matched baseline and intervened generations using the same prompt and initial noise for SD3.5 in Figs. C.4 and C.5 and for FLUX.1-dev in Figs. C.6 and C.7 on T2I-CompBench++. The examples include successful corrections of attribute swaps, attribute neglect, and entity neglect, together with cases in which steering changes the failure mode without producing a fully successful generation.

These examples complement the transition analysis in Sec. 6.3: stabilizing the assignment geometry can repair several downstream failure modes, but preserving the requested relation does not guarantee preservation of every constituent concept.

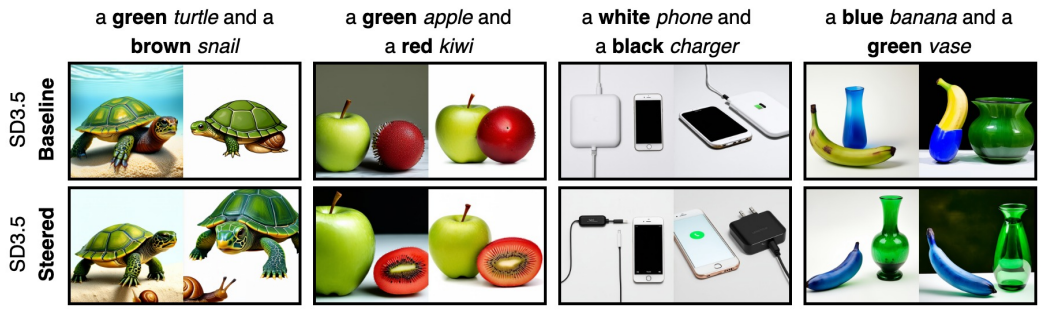

Figure C.4: **Additional qualitative intervention results for SD3.5 on T2I-CompBench++.** Each column compares the baseline and steered generation for the same prompt and initial noise. The intervention recovers the requested attribute-object correspondence in several failure cases while largely preserving object identity and scene structure.

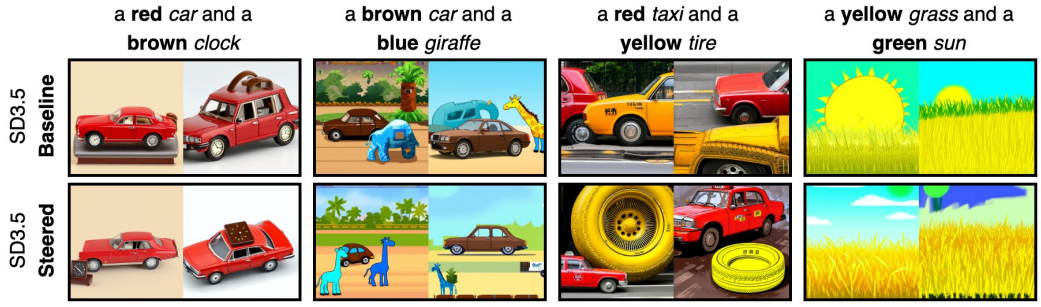

Figure C.5: **Additional qualitative intervention results for SD3.5 on T2I-CompBench++.** Each column compares the baseline and steered generation for the same prompt and initial noise. The intervention recovers the requested attribute-object correspondence in several failure cases while largely preserving object identity and scene structure.

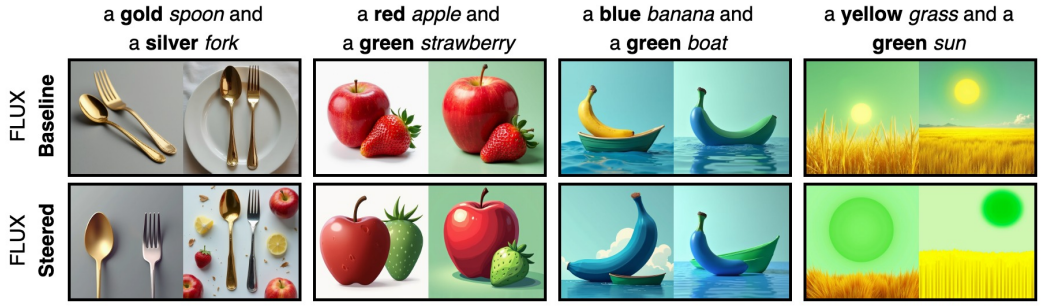

Figure C.6: **Additional qualitative intervention results for FLUX.1-dev on T2I-CompBench++.** Each column compares the baseline and steered generation for the same prompt and initial noise. The intervention recovers the requested attribute-object correspondence in several failure cases while largely preserving object identity and scene structure.

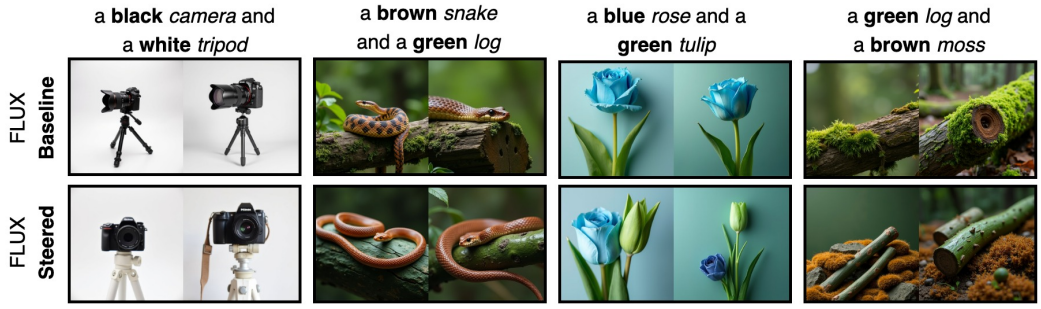

Figure C.7: **Additional qualitative intervention results for FLUX.1-dev on T2I-CompBench++.** Each column compares the baseline and steered generation for the same prompt and initial noise. The intervention recovers the requested attribute-object correspondence in several failure cases while largely preserving object identity and scene structure.

D LIMITATIONS

Automated evaluation. Our behavioral evaluation relies on Qwen3-8B to determine object presence, identity, attribute scorability, and the realized attribute-object correspondence. While this enables large-scale and structured evaluation, VQA-based judgments should still be interpreted with some caution. Importantly, ambiguity does not arise only from the evaluator itself: some generations exhibit mixed or hybridized object identities, where visual properties of one entity are incorporated into another, for example, a broccoli with strawberry-like texture or surface patterns. Such cases make a discrete assignment of object identity and attribute inherently ambiguous, even when both constituent concepts are visibly present. Representative examples are shown in Fig. D.1. We therefore treat the reported outcome frequencies as estimates under a fixed automated evaluation protocol rather than exact ground-truth annotations.

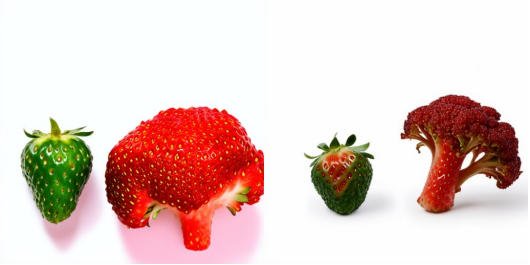

Figure D.1: **Ambiguous object identity under compositional failure.** Two SD3.5 generations for the prompt "a green strawberry and a red broccoli" exhibit mixed object identities, with visual properties of the strawberry appearing on the broccoli. Such hybridized generations make discrete object-identity and attribute assignments inherently ambiguous for automated evaluation.

Scope of the controlled benchmark. Our benchmark is deliberately diagnostic rather than comprehensive. We use minimal two-object prompts and focus on colour-object binding, allowing the same constituents and competing assignments to be controlled precisely. Moreover, the main Controlled Benchmark retains only object-attribute pairs that can be generated reliably in isolation. This design helps isolate composition-specific failures, but it does not reflect the full distribution of open-ended prompts, where difficult objects, longer syntax, spatial relations, counting, and other attribute types may interact. Extending the analysis to such settings is an important direction for future work.

Scope of the mechanistic readout. Our assignment geometry provides a compact representation-level description of how the two nouns are organized relative to the two attributes. The strong outcome-dependent trajectories and directional intervention controls show that this geometry is functionally actionable, but they do not establish that it is the unique mechanism responsible for compositional binding. Other representational directions, cross-modal interactions, and image-stream states may contribute to the final correspondence and are not exhaustively characterized by our current readouts.

Intervention side effects and model dependence. Steering the assignment geometry does not guarantee a fully successful final generation. Some trajectories corrected at the relational level transition instead to entity neglect or attribute neglect, indicating that preserving the requested correspondence and preserving all constituent content are related but distinct objectives. The intervention is also model dependent: while it produces large gains for SD3.5 on the Controlled Benchmark, the corresponding improvements are substantially smaller for FLUX.1-dev. The detector states, thresholds, steering schedule, and optimization strength are therefore backbone-specific and may require recalibration for other architectures, sampling procedures, or generation settings.