Token-to-Token Alignment of Text Embeddings for Semantic Blending

Saar Huberman[1,2] **Ron Mokady**[2] **Or Patashnik**[1] **Daniel Cohen-Or**[1]
[1]Tel Aviv University [2]BRIA AI

https://saarhuberman.github.io/token-to-token/

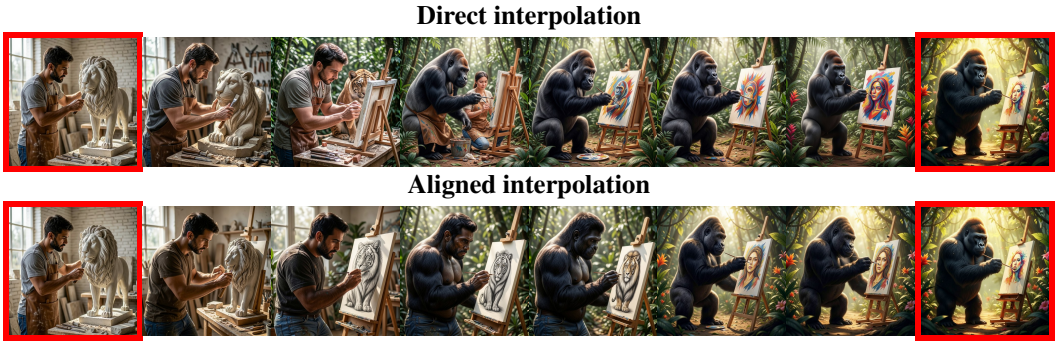


Figure 1: **Semantic interpolation through aligned text embeddings.** Direct interpolation between prompt embeddings yields inconsistent and entangled transitions. By aligning prompts at both the structural and embedding levels, we enforce token-wise semantic correspondence, enabling linear interpolation to follow a meaningful path in the text embedding space. This reveals a continuous semantic manifold that diffusion models can traverse to generate smooth and consistent visual transformations.

Abstract

In modern generative models, images are specified and controlled through text prompts. In practice, images are generated from sequences of tokens derived from these prompts. However, the space of token sequences lacks a consistent accessible structure: semantically similar images may correspond to sequences that differ in wording, ordering, and placement of concepts, while similar token sequences may encode very different semantics. This apparent lack of structure makes it difficult to perform smooth transitions in this space, hindering applications such as image blending and continuous control of edits. We argue that this limitation stems not from the absence of semantic structure, but from misalignment between representations. To address this misalignment, we introduce Token-to-Token alignment, a framework that establishes explicit semantic correspondence between tokens across prompts. Our approach transforms prompts into a structured representation in which semantically corresponding concepts are mapped to consistent positions across prompts, and then aligns their token embeddings based on semantic similarity. Concretely, the method consists of two stages: a structural alignment that rephrases prompts into a shared structured form, followed by an embedding-level alignment that matches token representations across prompts. With this alignment in place, simple linear interpolation becomes a meaningful operation, producing smooth and coherent semantic transitions and enabling applications such as blending and continuous editing. Our results show that text embedding spaces in text-to-image models implicitly encode a continuous semantic structure that becomes accessible once representations are properly aligned, suggesting that semantic control can be achieved by organizing existing representations rather than modifying the generative model.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Text-to-image generative models [46, 45, 24] operate on sequences of text tokens, whose embeddings define the semantic conditioning of the generation process. While this representation is highly expressive, it lacks a consistent structure that supports continuous manipulation. In particular, linear interpolation between prompt embeddings – an otherwise natural operation in latent spaces [49, 21] – often produces incoherent intermediate states. This limits the ability to perform smooth semantic transitions, such as blending concepts or gradually applying edits.

This issue is not merely technical, but structural. For example, the prompts "A man holding a ginger cat" and "A ginger cat is being held by a man" describe essentially the same scene, yet their tokenizations and embeddings differ significantly. When interpolated with a third prompt, such as "A man holding a black cat", these differences lead to inconsistent and entangled transitions. In this work, we argue that this limitation does not stem from the absence of semantic structure in the embedding space, but rather from misalignment between representations. Although text encoders implicitly organize semantics in a continuous manner, this structure is not directly accessible: corresponding concepts are not consistently represented across prompts. Without explicit correspondence, interpolation mixes unrelated attributes instead of following a meaningful semantic trajectory (see Figure 1).

To address this, we introduce *Token-to-Token alignment*, a framework that establishes explicit semantic correspondence between tokens across prompts. Our approach operates in two stages. First, we perform *structural alignment*, transforming prompts into a shared structured representation in which corresponding concepts occupy consistent semantic roles and textual positions. Second, we perform *embedding-level alignment*, matching token representations based on semantic similarity to resolve discrepancies in the text encoder space. Once representations are aligned, simple linear interpolation becomes a meaningful operation. Interpolating between aligned token embeddings produces coherent semantic trajectories, which the generative model renders as smooth and consistent visual transitions. This enables a unified framework for applications such as continuous image editing, and semantic blending, as demonstrated in Figure 2.

More broadly, our results suggest that semantic control in text-to-image models is fundamentally a problem of representation rather than manipulation. The text embedding space already encodes a continuous semantic structure, but this structure is entangled and inaccessible under standard prompt formulations. By enforcing token-level correspondence, we effectively linearize this space, turning interpolation from an unreliable heuristic into a principled operation. In this view, the generative model plays a largely passive role, acting as a renderer of trajectories defined in an aligned semantic space.

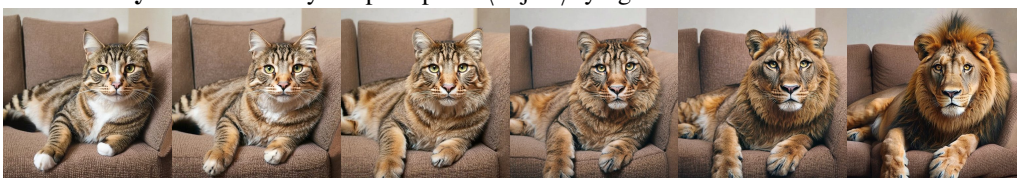
We validate our approach through extensive qualitative and quantitative evaluations across both continuous editing and continuous semantic blending tasks. Our method produces semantically coherent intermediate states at every point along the trajectory, outperforming prior approaches in consistency while maintaining competitive smoothness and visual quality. Ablation studies confirm the complementary roles of structural and embedding-level alignment, and user studies further demonstrate a clear preference for the trajectories generated by our method.

2 Related Work

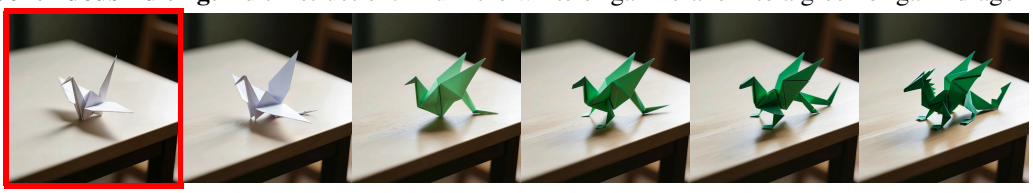
Structured Text Representations Recent advances in text-to-image diffusion models have shifted from short, underspecified prompts toward richer, more detailed descriptions [6, 25, 37, 20, 36, 29]. While alignment between short prompts is often trivial, underspecified prompts can still induce entangled priors [27], and the increased complexity and variability of richer descriptions introduce challenges in establishing correspondence across prompts. In natural language processing, canonicalization and structure-aware paraphrasing reduce such variability [5, 26], but operate on individual sentences rather than enforcing shared structure across different prompts. In contrast, we jointly transform prompt pairs into a shared structured form, enabling explicit token-level alignment.

More broadly, alignment of representation spaces has been studied across domains, including sequence matching and embedding alignment, using methods such as dynamic time warping [48], optimal transport [44], and cross-lingual embedding alignment [41, 12, 1]. Our setting involves embeddings from a shared text encoder with similar structure, where differences arise from both intended semantic changes (e.g., "cat" vs. "lion") and local variations in phrasing, tokenization, and contextualization, rather than global modality gaps.

Continuous Synthesis: modify the prompt "A ⟨object⟩ lying on the couch" from cat→lion



Continuous Editing: Edit instruction: "Turn the white origami crane into a green origami dragon"



Continuous Blending: generate intermediate scenes



Figure 2: **Applications enabled by aligned text embedding interpolation.** Once representations are aligned, simple linear interpolation produces coherent semantic transitions. This unified framework supports continuous synthesis (top), image editing (middle), and image blending (bottom), generating meaningful intermediate states across tasks. Red boxes denote input images.

Latent Space Interpolation Latent spaces in generative models often exhibit meaningful structure, enabling simple operations such as linear interpolation to produce smooth and semantically coherent transitions between images [49, 21, 43]. In text-conditioned diffusion models, semantic manipulation has been explored across multiple representation spaces, including noisy latent trajectories [39, 34, 14], intermediate features such as attention maps [22, 50, 38], and parameter-space modifications such as LoRA-based methods [18, 16, 57, 10]. Closest to our work are methods operating in text embedding and contextual representation spaces, including approaches that decompose text embeddings into interpretable components [30, 4, 56], perform per-edit optimization [32, 52], operate in global embedding spaces [15, 19], or directly modify textual representations [17, 13]. In contrast, we align prompt pairs at the token level, establishing explicit semantic correspondence across their representations.

Image Editing, Blending and Continuous Control Early diffusion-based editing methods rely on text prompts as the primary control mechanism, but this interface mainly supports discrete changes and provides limited control over transformation strength [39, 22, 3, 7]. A growing body of recent work focuses on continuous control, introducing mechanisms to modulate edit strength. These mechanisms include guidance-based control [51, 59], LoRA-based editing directions [57, 18], and fine-tuned instruction-based editors with explicit strength scalars [42, 53, 38]. Related work has also leveraged video generation for image editing [47] and analyzed and smoothed semantic progression in generated sequences [40].

Continuous transitions have also been explored through pixel or latent-space interpolation [8, 58], but often produce visual blends with artifacts such as ghosting or inconsistent object composition, rather than semantically meaningful intermediate states. More recent work blends distinct concepts by identifying shared semantic attributes [55]. However, these approaches do not explicitly enforce structured correspondence between concepts, which can lead to intermediate states where semantic attributes are not consistently preserved.

In contrast, our approach enables structured, continuous semantic transitions directly in the text conditioning space, without task-specific training or modifications to the generative model. Moreover, it generalizes to both spatially similar and spatially distinct images.

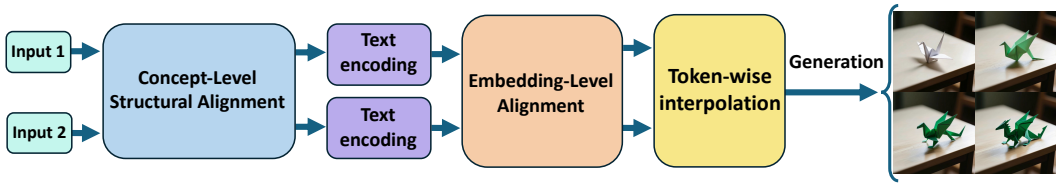


Figure 3: **Method overview.** Inputs are transformed into structured descriptions under a shared schema with aligned textual phrasing, establishing correspondence between scene components and how they are expressed. Their token embeddings are then aligned in the text encoder space to enforce token-wise semantic correspondence. Interpolation between aligned embeddings produces intermediate representations, which the generative model renders as smooth semantic transitions.

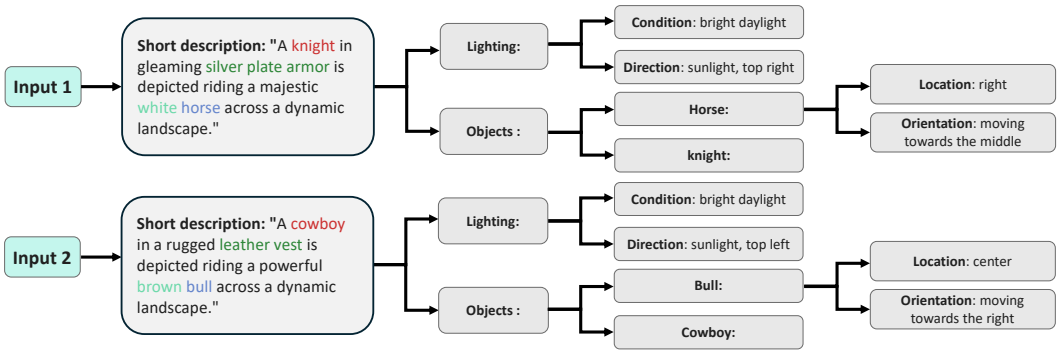


Figure 4: **Concept level structural alignment.** Inputs are transformed into structured descriptions under a shared schema, where scene components are decomposed into semantic fields (e.g., lighting, objects) with consistent ordering. Corresponding elements are expressed using aligned textual phrasing within each field, establishing coarse correspondence at the textual level.

3 Method

In this section, we present our framework for continuous semantic control in image generation and editing. At the core of our framework is a Token-to-Token alignment method that receives as input a pair of natural language text prompts p^A, p^B, and produces a corresponding pair of aligned token embedding sequences E^A, E^B. These aligned sequences establish semantic correspondences across tokens, enabling meaningful interpolation between the prompts.

Given the aligned embedding sequences, E^A and E^B, we interpolate between them to obtain intermediate representations that capture gradual semantic transitions. These interpolated embeddings are then used to condition a generative model G, producing a sequence of images along the transition. The generative model is a diffusion model, which takes Gaussian noise as input and progressively denoises it while being conditioned on a text embedding. In our case, this conditioning is provided by the interpolated embeddings produced by our alignment method. For applications that require an input image, we use diffusion models that additionally support image conditioning alongside the text embedding. Specifically, we demonstrate our framework using FIBO [20] and Flux2 [35]. FIBO has two variants: one conditioned solely on a text prompt, and another that additionally supports image conditioning. In Flux2, the same model can be used with or without image conditioning.

Next, we present our Token-to-Token method, and then in Section 3.2 we describe how it is applied to different generative applications. An overview of our method is shown in Figure 3.

3.1 Token-to-Token

Concept-Level Structural Alignment The first stage of Token-to-Token establishes concept-level structural alignment using an LLM. Given a pair of prompts p^A, p^B, we instruct the LLM to jointly construct aligned scene descriptions J^A, J^B under explicit structural constraints. The goal of this stage is to reduce linguistic variability and enforce consistent semantic structure prior to tokenization.

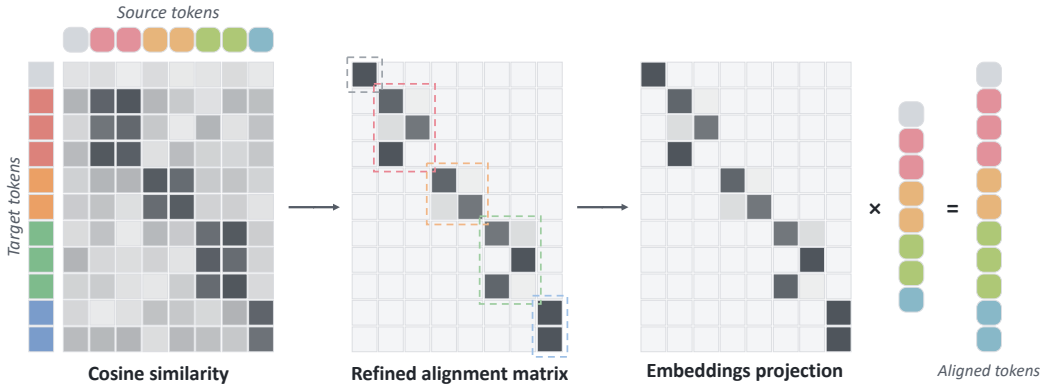


Figure 5: **Embedding-level alignment.** We compute a pairwise cosine similarity matrix between token embeddings, then refine it using field-based masking and a locality bias, and apply a temperature-scaled softmax to obtain alignment weights. These weights define a projection that maps source embeddings onto the target token structure, re-expressing their semantic content in the target layout and establishing token-wise semantic correspondence.

We illustrate a pair of aligned scene descriptions in Figure 4. Each description is represented as a JSON string with the same fixed semantic fields (e.g., objects, attributes, lighting) and consistent ordering. Corresponding elements use parallel phrasing within each field, such that semantic differences appear as local substitutions within a shared context rather than independent rephrasing (e.g., *knight* vs. *cowboy*, *horse* vs. *bull*), highlighted by matching colors in the figure.

While these descriptions establish concept-level alignment, they operate prior to tokenization and do not guarantee correspondence in the text encoder space. Semantically corresponding words (e.g., *knight* vs. *cowboy*) may tokenize into different numbers of tokens, shifting token positions and misaligning the resulting embedding sequences.

Embedding-Level Alignment Let $V^A = \{v_i^A\}_{i=1}^n$ and $V^B = \{v_j^B\}_{j=1}^m$ denote the token embeddings produced by the text encoder for J^A and J^B, respectively. Due to differences in phrasing and tokenization, these sequences may differ in length and token boundaries, even after structural alignment.

To address this, we perform embedding-level alignment to establish token correspondences across the two sequences, as illustrated in Figure 5. We first compute a pairwise similarity matrix $S \in \mathbb{R}^{n \times m}$ using cosine similarity:

$$S_{ij} = \frac{\langle v_i^A, v_j^B \rangle}{\|v_i^A\| \|v_j^B\|}.$$

Token embeddings from transformer-based text encoders are contextual and reflect semantic roles within the description, enabling similarity-based alignment rather than exact token matching, even when corresponding elements differ in value (e.g., colors, clothing, or expressions).

Alignment is constrained by the semantic structure obtained in the previous stage. Specifically, we restrict matching to tokens within corresponding semantic fields by masking S, ensuring that correspondence respects the scene decomposition. To encourage stable correspondences, we incorporate a locality bias over token positions and apply temperature scaling: $\tilde{S}_{ij} = S_{ij}/\tau + \lambda \cdot \mathrm{pos}(i, j)$, where $\mathrm{pos}(i, j)$ is a Gaussian bias over relative token positions within each semantic field. Following structural alignment, corresponding tokens are likely to appear at similar relative positions, motivating the locality bias. Temperature scaling amplifies similarity differences before normalization, preventing correspondence weights from spreading across many unrelated tokens.

We obtain a soft alignment matrix A by normalizing similarities over tokens in V^A, $A_{ij} = \exp(\tilde{S}_{ij}) / \sum_{i'} \exp(\tilde{S}_{i'j})$. For each token in V^B, this defines a soft correspondence over tokens in V^A, allowing sequences of different lengths to be matched. We then project the semantic content of V^A onto the structure of V^B, yielding $E^A = A^\top V^A$ and $E^B = V^B$, where $\top$ denotes transpose.

Interpolation in Aligned Embedding Space Given the aligned token embedding sequences $E^A = \{e_j^A\}_{j=1}^m$ and $E^B = \{e_j^B\}_{j=1}^m$, we interpolate between them by linearly combining corresponding embeddings:

$$E^\alpha = \{(1-\alpha)e_j^A + \alpha e_j^B\}_{j=1}^m, \quad \alpha \in [0,1].$$

Crucially, interpolation is performed between semantically corresponding tokens, as established by the alignment stages, ensuring that intermediate representations reflect gradual and coherent semantic changes rather than mixtures of unrelated attributes. These interpolated embeddings can be directly used as conditioning for the generative model, which renders them as smooth semantic transitions in the image space.

3.2 Applications

Natural language can specify the direction of a semantic change, but offers limited control over its magnitude. The interpolation coefficient α provides continuous control over the transformation strength. We next describe how Token-to-Token enables continuous synthesis, editing, and blending.

Continuous Synthesis In this application, shown in the top row of Figure 2, the input consists of a pair of prompts p^A, p^B that differ in a specific attribute (e.g., the object on the couch). We apply Token-to-Token to construct a sequence of interpolated embeddings $\{E^\alpha\}_{\alpha\in[0,1]}$, and generate the corresponding images $\{G(E^\alpha)\}_{\alpha\in[0,1]}$ using text-only conditioning.

Continuous Editing In this application, shown in the middle row of Figure 2, the input consists of an image I and a text prompt p^B describing the desired edit. We first caption the input image to obtain a source prompt p^A, and then apply Token-to-Token to align p^A with the target prompt p^B and construct a sequence of interpolated embeddings $\{E^\alpha\}_{\alpha\in[0,1]}$. We generate the corresponding images $\{G(I, E^\alpha)\}_{\alpha\in[0,1]}$ by conditioning the model on both the input image and each interpolated embedding. In this setting, we use Flux2 and the image-conditioned variant of FIBO. This produces a continuous transition from the input image toward the edited result, enabling gradual control over the edit strength.

Continuous Blending In this application, shown in the bottom row of Figure 2, the input consists of a pair of images I^A, I^B. We caption the images to obtain a pair of prompts p^A, p^B, and apply Token-to-Token to construct a sequence of interpolated embeddings $\{E^\alpha\}_{\alpha\in[0,1]}$. We generate the corresponding images $\{G(I^A, I^B, E^\alpha)\}_{\alpha\in[0,1]}$ by conditioning the model on both input images and each interpolated embedding. To incorporate both input images, their embeddings are concatenated in the model input and distinguished using separate temporal positional encodings, allowing the model to identify their respective roles. The interaction between the generated image, the input images, and the text embeddings is handled through the attention layers of the diffusion transformer, where tokens attend to each other to exchange information.

A key challenge in this application is balancing the contribution of the two input images throughout the interpolation. Naively conditioning on both images can lead to dominance of visual features, reducing the process to appearance-based blending. To address this, we modulate the attention weights using an α-dependent scaling, weighting the contributions of I^A and I^B by $(1-\alpha)^2$ and α^2, respectively. This emphasizes image conditioning near the endpoints to preserve fidelity, while reducing it in intermediate regions to better follow the interpolated semantic trajectory. This results in coherent blending, where semantic attributes evolve smoothly while remaining visually grounded. In this setting, we use Flux2 and the image-conditioned variant of FIBO.

4 Experiments and Results

We evaluate our framework on both applications through qualitative (Section 4.1) and quantitative (Section 4.2) comparisons, and conduct ablation studies (Section 4.3) to assess the contribution of key components.

Implementation Details. We evaluate our method using FLUX-2 (Klein 9B) [35] and FIBO-edit [20], with Qwen3 (8B) [54] and SmolLM3-3B [2] as their corresponding text encoders. Structured scene descriptions are generated using Gemini 2.5 [11] under a shared JSON schema. For embedding alignment, we use intermediate text representations from multiple encoder layers and compute layer-wise similarity matrices with temperature scaling ($\tau = 0.05$). For continuous blending, image contributions are modulated using α-dependent quadratic weighting. Unless otherwise specified, inference settings are kept consistent across methods for fair comparison.

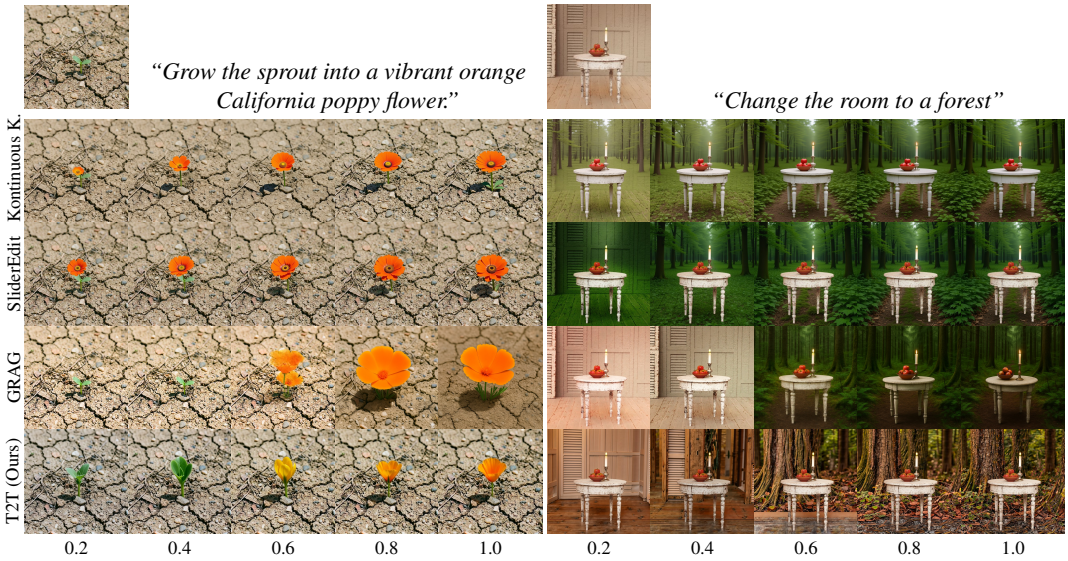


Figure 6: **Qualitative comparison with continuous editing methods.** Existing methods often rely on appearance-based interpolation or introduce abrupt semantic changes, whereas our method produces smooth and semantically coherent transitions with meaningful intermediate states.



Figure 7: **Qualitative comparison with continuous blending methods.** While prior approaches may produce mixed or drifting attributes along the transition, our method interpolates in aligned text embedding space, yielding coherent and semantically valid intermediate images at every step.

Baselines. We compare our method against prior work across two settings. For continuous image blending, we consider DiffMorpher [58], a diffusion-based method that morphs between images by interpolating LoRA representations; FreeMorph [8], a tuning-free approach that blends attention features; and VibeSpace [55], which learns a joint embedding space and performs latent interpolation. For continuous image editing, we consider Kontinuous Kontext [42], a supervised method that learns a scalar control for edit strength; SliderEdit [57], which enables continuous control via token-selective LoRA adapters; and GRAG [59], a training-free method that modulates attention representations. All baselines are evaluated using their official implementations and recommended settings.

4.1 Qualitative Results

We present qualitative evaluations for continuous editing in Figure 6 and continuous blending in Figure 7. As shown in Figure 6, prior methods often rely on appearance-based transitions or introduce abrupt semantic changes during the edit trajectory. For example, in the left example,

Table 1: Quantitative evaluation with existing continuous blending methods.

Method	Morph4Data			BlendBench		
	PPL ↓	MUSIQ ↑	FID ↓	PPL ↓	MUSIQ ↑	FID ↓
DiffMorpher	**12.40**	62.27	113.68	**21.54**	65.95	146.80
FreeMorph	27.34	67.51	105.85	63.06	71.17	129.53
Vibe Space	61.96	71.04	101.76	252.47	71.53	139.60
$T2T_{Flux2\text{-}klein}$	45.38	**74.61**	105.97	147.63	**73.27**	**85.06**
$T2T_{FIBO\text{-}edit}$	53.49	73.06	**85.71**	164.71	73.08	103.64

Table 2: Quantitative evaluation with existing continuous editing methods.

Method	PIE-Bench		
	δ_{smooth} ↓	Continuity↑	Norm CLIP-Dir↑
Kontinuous K.	**0.246**	1.33	0.259
Slider Edit	0.395	1.08	0.353
GRAG	0.515	0.97	0.343
$T2T_{Flux2\text{-}klein}$	0.562	1.20	0.355
$T2T_{FIBO\text{-}edit}$	0.530	**1.46**	**0.393**

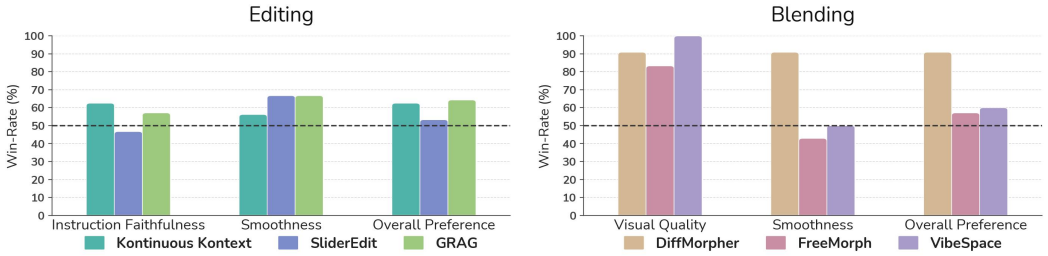


Figure 8: **User study results.** Win rates (%) of our method in pairwise comparisons against editing (left) and blending (right) baselines across evaluation criteria.

existing methods either abruptly generate the flower or mainly modify its scale without modeling a meaningful semantic growth process. In contrast, our method produces smooth semantic transitions with coherent intermediate states that progressively reflect the intended edit. As shown in Figure 7, methods that operate in pixel or latent spaces (e.g., DiffMorpher, FreeMorph) often generate distorted intermediate images, particularly under large geometric or viewpoint changes. DiffMorpher exhibits ghosting artifacts with overlapping structures, while FreeMorph may produce implausible body configurations with structural discontinuities. Although VibeSpace operates in a semantic space, it still exhibits attribute drift and less consistent transitions across intermediate results. In contrast, our method produces coherent intermediate images by interpolating between semantically aligned text embeddings, preserving structure and ensuring that each step corresponds to a valid scene. Additional editing examples and blending transitions are provided in the supplementary material.

4.2 Quantitative Results

Continuous Blending. We evaluate continuous blending on **Morph4Data** [8] and our proposed **BlendBench**. Morph4Data contains 76 image pairs with partial spatial alignment, enabling blending through local correspondence. To evaluate semantic blending in more challenging settings, we introduce BlendBench, a dataset of 100 complex pairs with variations in objects, poses, and viewpoints, requiring simultaneous semantic changes. These scenarios lack reliable spatial correspondence and highlight limitations of morphing-based methods, such as ghosting and structural artifacts, making blending more challenging. We further quantify this distinction in Appendix A.1. We follow [8] using 5 intermediate steps for Morph4Data, and use 9 steps for BlendBench to capture more complex transitions. We evaluate blending in terms of **trajectory smoothness**, measured using PPL [31] and **visual quality**, measured using MUSIQ [33] and FID computed against the input image distribution [23] (see supplementary material for details).

As shown in Table 1, morphing-based methods (DiffMorpher, FreeMorph) achieve the lowest PPL, indicating smooth perceptual trajectories. However, PPL captures local perceptual smoothness and does not distinguish a semantically meaningful transition from a visually smooth blend; such trajectories may therefore achieve low PPL while exhibiting ghosting or implausible intermediate states. The lower MUSIQ and higher FID further reflect this degradation in visual quality. VibeSpace improves semantic consistency but exhibits less stable trajectories and weaker adherence to the input images. In contrast, our method produces more grounded and structurally consistent transitions while maintaining competitive smoothness, with higher image quality and better alignment with the input distribution.

Continuous Editing. Following the evaluation protocol of Kontinuous Kontext [42], we evaluate continuous editing on a subset of PIE-Bench [28] containing 540 image-instruction pairs with 6 edit strengths. We measure **smoothness** using δ_{smooth} [42], **semantic alignment** using normalized

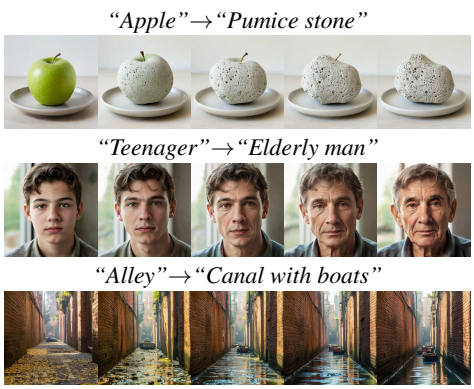


Figure 9: **Continuous synthesis using Token-to-Token alignment.** Interpolating between aligned text embeddings produces smooth semantic transitions while preserving structural consistency throughout the trajectory.



Figure 10: **Qualitative alignment ablation study.** Direct interpolation produces abrupt transitions. Text-level alignment improves continuity but lacks fine-grained detail, while embedding alignment alone leads to unstable transformations. Combining both yields smooth and coherent transitions.

CLIP direction (Norm CLIP-Dir) [51], and **continuity** in CLIP space [57], which evaluates whether semantic changes evolve uniformly across edit strengths (see supplementary material for details).

As shown in Table 2, Kontinuous Kontext achieves the strongest smoothness, reflecting its design for perceptually consistent transitions, but produces trajectories that are less aligned with the intended semantic edit, as reflected by lower CLIP-Dir. SliderEdit and GRAG achieve stronger semantic alignment, but exhibit reduced continuity and less stable editing trajectories across edit strengths. In contrast, our method achieves strong semantic alignment together with improved continuity, producing trajectories that follow a coherent semantic direction while maintaining competitive smoothness.

Table 3: Quantitative alignment ablation study.

	PIE-Bench		
Method	δ_{smooth} ↓	**Continuity**↑	**Norm CLIP-Dir**↑
Direct	1.157	1.232	0.189
Text align	0.713	1.294	0.198
Embedding align	0.749	1.207	0.051
Full w/o locality bias	<u>0.678</u>	<u>1.417</u>	0.263
Full w/o temperature	9.498	0.751	<u>0.302</u>
Full method	**0.530**	**1.461**	**0.393**

User Study. We conduct a pairwise user study comparing our method against continuous blending and editing baselines. Participants evaluate transition sequences randomly sampled from the evaluation datasets based on semantic smoothness and overall preference for both tasks, visual quality for blending, and instruction faithfulness for editing (see Fig. 11). In total, we collect 140 assessments from 20 participants. As shown in Fig. 8, our method achieves higher win rates across most criteria, particularly in visual quality and overall preference for blending, while also improving semantic smoothness and overall preference for editing.

4.3 Ablation Studies

We evaluate the contribution of each alignment stage through qualitative (Fig. 10) and quantitative (Table 3) ablations. As shown in Fig. 10, direct interpolation produces abrupt transitions, text-level alignment improves continuity but lacks fine-grained detail, and embedding-level alignment alone remains unstable without the consistent semantic structure established by text-level alignment. Combining both stages yields smooth and semantically coherent transitions. Table 3 further supports these observations quantitatively, with the full method achieving the strongest performance across all metrics. Removing the locality bias degrades performance, confirming the benefit of exploiting the positional structure established by text-level alignment, while removing temperature scaling causes the correspondence weights to become overly diffuse, substantially degrading interpolation quality.

5 Conclusion

We introduced Token-to-Token alignment, a framework for enabling coherent semantic interpolation directly in the text embedding space. At a higher level, we suggest a perspective on semantic control in text-to-image generation as a problem of representation rather than manipulation. While the text embedding space implicitly encodes a meaningful and continuous semantic structure, this structure is not directly accessible due to misalignment between prompts.

By enforcing explicit token-level correspondence, we effectively linearize this space, transforming interpolation from an unreliable operation into a principled semantic operation. In this view, the generative model plays a largely passive role, acting as a renderer of trajectories defined in the aligned embedding space. This perspective shifts the emphasis away from model modification and toward organizing existing representations as the primary mechanism for control.

At the same time, our approach relies on the quality of the structural alignment between prompts. While embedding-level alignment can resolve local mismatches, it may fail when the initial descriptions do not establish the intended semantic correspondence. In addition, while linear interpolation becomes meaningful under alignment, it may not fully capture more complex or highly non-linear semantic transformations. These limitations point to several directions for future work, including improving the robustness of alignment, exploring richer and more flexible trajectories in the embedding space, and extending this formulation to temporally coherent settings such as video, where aligned semantic paths could provide a natural mechanism for controlling continuous evolution over time.

Acknowledgments

This research was supported in part by the Israel Science Foundation (grants no. 3441/21 and 1473/24), Len Blavatnik and the Blavatnik Family Foundation, and the Bronfman Family Foundation.

Competing Interests

The authors declare no competing interests.

References

[1] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, 2018.

[2] Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. `https://huggingface.co/blog/smollm3`, 2025.

[3] Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025.

[4] Stefan Andreas Baumann, Felix Krause, Michael Neumayr, Nick Stracke, Melvin Sevi, Vincent Tao Hu, and Björn Ommer. Continuous, subject-specific attribute control in t2i models by identifying semantic directions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13231–13241, 2025.

[5] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, 2014.

[6] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[8] Yukang Cao, Chenyang Si, Jinghao Wang, and Ziwei Liu. Freemorph: Tuning-free generalized image morphing with diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18111–18120, 2025.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Pin-Yen Chiu, I Fang, Jun-Cheng Chen, et al. Text slider: Efficient and plug-and-play continuous concept control for image/video synthesis via lora adapters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 613–622, 2026.

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[12] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[13] Omer Dahary, Benaya Koren, Daniel Garibi, and Daniel Cohen-Or. On-the-fly repulsion in the contextual space for rich diversity in diffusion transformers. *arXiv preprint arXiv:2603.28762*, 2026.

[14] Yusuf Dalva and Pinar Yanardag. Noiseclr: A contrastive learning approach for unsupervised discovery of interpretable directions in diffusion models, 2023.

[15] Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow transformers, 2024.

[16] Amil Dravid, Yossi Gandelsman, Kuan-Chieh Wang, Rameen Abdal, Gordon Wetzstein, Alexei A Efros, and Kfir Aberman. Interpreting the weight space of customized diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[17] Yigit Ekin and Yossi Gandelsman. The unreasonable effectiveness of text embedding interpolation for continuous image steering. *arXiv preprint arXiv:2603.17998*, 2026.

[18] Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pages 172–188. Springer, 2024.

[19] Daniel Garibi, Shahar Yadin, Roni Paiss, Omer Tov, Shiran Zada, Ariel Ephrat, Tomer Michaeli, Inbar Mosseri, and Tali Dekel. Tokenverse: Versatile multi-concept personalization in token modulation space, 2025. URL `https://arxiv.org/abs/2501.12224`.

[20] Eyal Gutflaish, Eliran Kachlon, Hezi Zisman, Tal Hacham, Nimrod Sarid, Alexander Visheratin, Saar Huberman, Gal Davidi, Guy Bukchin, Kfir Goldberg, and Ron Mokady. Generating an image from 1,000 words: Enhancing text-to-image with structured captions. *arXiv preprint arXiv:2511.06876*, abs/2511.06876, 2025.

[21] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020.

[22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[23] 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. *Advances in neural information processing systems*, 30, 2017.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[25] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[26] James Y Huang, Kuan-Hao Huang, and Kai-Wei Chang. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1372–1379, 2021.

[27] Saar Huberman, Or Patashnik, Omer Dahary, Ron Mokady, and Daniel Cohen-Or. Image generation from contextually-contradictory prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14885–14894, 2026.

[28] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] Eliran Kachlon, Alexander Visheratin, Nimrod Sarid, Tal Hacham, Eyal Gutflaish, Saar Huberman, Hezi Zisman, David Ruppin, and Ron Mokady. Bbq-to-image: Numeric bounding box and qolor control in large-scale text-to-image models. *arXiv preprint arXiv:2602.20672*, 2026.

[30] Ronen Kamenetsky, Sara Dorfman, Daniel Garibi, Roni Paiss, Or Patashnik, and Daniel Cohen-Or. Saedit: Token-level control for continuous image editing via sparse autoencoder. *arXiv preprint arXiv:2510.05081*, 2025.

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[32] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

[33] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[34] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19721–19730, 2025.

[35] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

[36] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024.

[37] Mushui Liu, Yuhang Ma, Zhen Yang, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. Llm4gen: Leveraging semantic representation of llms for text-to-image generation. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 39, pages 5523–5531, 2025.

[38] Zhixuan Liu, Peter Schaldenbrand, Yijun Li, Long Mai, Aniruddha Mahapatra, Cusuh Ham, Jean Oh, and Jui-Hsien Wang. Tokendial: Continuous attribute control in text-to-video via spatiotemporal token offsets, 2026. URL `https://arxiv.org/abs/2603.27520`.

[39] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[40] Gal Metzer, Sagi Polaczek, Ali Mahdavi-Amiri, Raja Giryes, and Daniel Cohen-Or. Video analysis and generation via a semantic progress function. *arXiv preprint arXiv:2604.22554*, 2026.

[41] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[42] Rishubh Parihar, Or Patashnik, Daniil Ostashev, R Venkatesh Babu, Daniel Cohen-Or, and Kuan-Chieh Wang. Kontinuous kontext: Continuous strength control for instruction-based image editing. *arXiv preprint arXiv:2510.08532*, 2025.

[43] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2085–2094, 2021.

[44] Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. URL `https://arxiv.org/abs/2102.12092`.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[47] Noam Rotstein, Gal Yona, Daniel Silver, Roy Velich, David Bensaïd, and Ron Kimmel. Pathways on the image manifold: Image editing via video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7857–7866, 2025.

[48] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[49] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 44(4): 2004–2018, 2020.

[50] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1921–1930, 2023.

[51] Alon Wolf, Chen Katzir, Kfir Aberman, and Or Patashnik. Continuous control of editing models via adaptive-origin guidance. *arXiv preprint arXiv:2602.03826*, 2026.

[52] 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*, pages 1900–1910, 2023.

[53] Zhenyu Xu, Xiaoqi Shen, Haotian Nan, and Xinyu Zhang. Numerikontrol: Adding numeric control to diffusion transformers for instruction-based image editing. *arXiv preprint arXiv:2511.23105*, 2025.

[54] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[55] Huzheng Yang, Katherine Xu, Andrew Lu, Michael D Grossberg, Yutong Bai, and Jianbo Shi. Vibe spaces for creatively connecting and expressing visual concepts. *arXiv preprint arXiv:2512.14884*, 2025.

[56] Hu Yu, Hao Luo, Fan Wang, and Feng Zhao. Uncovering the text embedding in text-to-image diffusion models. *arXiv preprint arXiv:2404.01154*, 2024.

[57] Arman Zarei, Samyadeep Basu, Mobina Pournemat, Sayan Nag, Ryan Rossi, and Soheil Feizi. Slideredit: Continuous image editing with fine-grained instruction control. *arXiv preprint arXiv:2511.09715*, 2025.

[58] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. Diffmorpher: Unleashing the capability of diffusion models for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2024.

[59] Xuanpu Zhang, Xuesong Niu, Ruidong Chen, Dan Song, Jianhao Zeng, Penghui Du, Haoxiang Cao, Kai Wu, and An-an Liu. Group relative attention guidance for image editing. *arXiv preprint arXiv:2510.24657*, 2025.

A Additional details

A.1 Benchmarks

Morph4data Following the protocol of FreeMorph [8], we evaluate semantic interpolation on Morph4Data, a curated dataset of 76 image pairs designed for morphing tasks. The dataset is organized into four categories: (i) similar layout but different semantics, (ii) similar layout and semantics (including face images from CelebA-HQ and object categories such as cars), (iii) dissimilar layout and semantics sampled from ImageNet, and (iv) real-world animal images (e.g., cats and dogs) collected from the internet. Each pair defines a source and target image, and we generate intermediate samples using uniformly spaced interpolation coefficients. Following the original setup, we use 5 interpolation steps between endpoints.

BlendBench We introduce BlendBench, a benchmark for evaluating continuous semantic interpolation under correspondence defined by object interactions. The dataset consists of 100 image pairs constructed by first generating pairs of prompts that describe distinct scenes involving multiple objects participating in an action. For each pair, prompts are designed such that objects occupy analogous semantic roles within the interaction, while varying object identity, attributes, background, and spatial configuration (e.g., "a samurai standing and holding a sword in the middle of a forest" vs.

"a tennis player running and hitting a ball with a racket"). This construction ensures that semantic correspondence is defined by the roles of objects within the interaction, rather than by shared appearance or spatial alignment. Each prompt is then independently used to generate an image using Gemini 2.5 [11], and the resulting images form the source and target pair.

For each pair, methods are evaluated on 9 uniformly spaced intermediate samples capturing the semantic transition.

Quantifying Differences Between Blending Benchmarks. To quantify the distinction between the two blending benchmarks, we compare their scene complexity and spatial correspondence in Table 4. Using an off-the-shelf DETR-ResNet-50 [9] detector with a confidence threshold of 0.7, we measure the number of detected entities per image and the spatial consistency of the primary subject across endpoints.

Table 4: Quantitative comparison of Morph4Data and BlendBench.

Metric	**Morph4Data**	**BlendBench**
Entities per image (mean)	1.9	7.4
Single-entity images	62%	21%
Primary-subject box IoU (median)	0.72	0.38
Primary-subject centroid shift (median, fraction of frame)	0.06	0.14

As shown in Table 4, BlendBench contains substantially more entities per image and weaker spatial correspondence between endpoints than Morph4Data. These results show that Morph4Data primarily evaluates transitions under partial spatial alignment, whereas BlendBench emphasizes semantic correspondence across more complex, spatially misaligned scenes.

Simplified PIE-Bench. We follow the evaluation protocol of Kontinuous Kontext [42] and construct a simplified subset of PIE-Bench [28] for evaluating continuous semantic interpolation. For consistency with prior work and to enable fair comparison, we exclude the roughness, transparency, and style categories, resulting in 540 image–instruction pairs.

Because many original instructions involve multiple simultaneous modifications, which are not well suited for evaluating gradual transitions, we simplify the instructions using an LLM. The model is prompted to rewrite each instruction so that it involves at most two attribute changes.

A.2 Metrics

Continuous Blending metrics. We evaluate continuous blending quality using PPL, FID and MUSIQ.

PPL. We compute the perceptual path length (PPL) following [31], using LPIPS distances between consecutive frames with a VGG backbone. Formally,

$$\text{PPL} = (N-1)^2 \sum_{i=1}^{N-1} \text{LPIPS}(I_{i+1}, I_i)^2.$$

This metric captures the consistency of changes along the trajectory, where lower values indicate more uniform transitions.

MUSIQ. We evaluate perceptual image quality using MUSIQ [33]. Scores are computed per frame and averaged across all intermediate steps and examples. Higher values indicate better visual quality.

FID. We evaluate distributional similarity using Fréchet Inception Distance (FID) [23]. Real images correspond to all source and target images across the dataset, while generated images correspond to all intermediate frames (excluding endpoints) pooled across all examples. Lower values indicate better alignment with the real image distribution.

Continuous editing metrics. We evaluate continuous editing behavior using δ_{smooth}, continuity, and normalized CLIP direction (Norm CLIP-Dir).

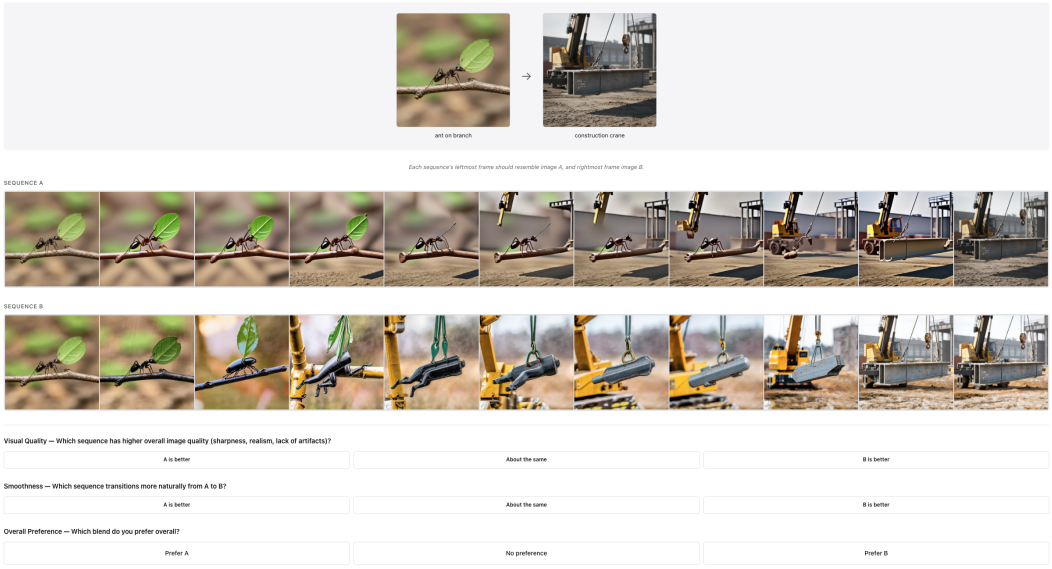


Figure 11: User study interface used for pairwise evaluation of continuous blending. Participants compare two transition sequences and select preferences based on visual quality, smoothness, and overall preference.

δ_{smooth}. We evaluate smoothness using the δ_{smooth} metric [42], which measures second-order consistency along the trajectory. It is defined based on the triangle inequality between consecutive triplets of frames, capturing deviations from linear transitions (see [42] for full details). Lower values indicate smoother local transitions.

Continuity. We measure continuity following [57] by evaluating the uniformity of similarity scores computed in CLIP embedding space along the editing trajectory. A chi-squared statistic is applied over binned similarity values to quantify deviations from a uniform progression (see [57] for full details), where higher values indicate smoother and more consistent transitions.

Norm CLIP-Dir. We measure text alignment consistency using the normalized CLIP direction metric [51], which evaluates whether each step follows the intended semantic edit direction. Specifically, we compute the cosine similarity between the stepwise image direction and the text direction in CLIP embedding space, normalized by the global image-text alignment:

$$\frac{1}{N-1}\sum_{i=1}^{N-1}\frac{\cos(\Delta\mathbf{v}_{img}^{(i)},\Delta\mathbf{v}_{text})}{\cos(\Delta\mathbf{v}_{img}^{(global)},\Delta\mathbf{v}_{text})},$$

where $\Delta\mathbf{v}_{img}^{(i)} = \text{CLIP}(I_{i+1}) - \text{CLIP}(I_i)$ is the local edit direction, $\Delta\mathbf{v}_{img}^{(global)} = \text{CLIP}(I_{N-1}) - \text{CLIP}(I_0)$ is the global edit direction, and $\Delta\mathbf{v}_{text} = \text{CLIP}(c_{edit}) - \text{CLIP}(c_{src})$ is the text direction. Higher values indicate better alignment with the intended edit.

A.3 User Study Details

We conduct separate pairwise comparison studies for continuous blending and continuous editing. Participants are presented with two transition sequences generated by our method and a baseline method, shown in randomized order.

For **continuous blending**, participants evaluate: (i) visual quality, (ii) smoothness of the transition, and (iii) overall preference.

For **continuous editing**, participants evaluate: (i) instruction faithfulness, (ii) smoothness, and (iii) overall preference.

Figure 11 shows the interface used for the continuous blending evaluation.

Stage 1. `get_paired_prompts_blend`**: joint generation from** (I_A, I_B)

```
You receive TWO images and produce TWO aligned JSONs (source, edit).

- Same keys, same #objects, same field order.
- Identical wording for fields not affected by the visual diff.
- Affected fields keep the same sentence structure; substitute only
  the minimum words required.
- edit.edit_instruction: imperative description of the change.
- source.edit_instruction: counter-instruction preserving the
  original (e.g., edit "Change hair to red" -> source "Keep hair brown").
```

Stage 2. `refine_alignment`**: image-aware bidirectional refinement**

```
Classify each field by word-diff ratio:

unaffected (<10%, no overlap w/ edit_instruction) -> force-copy
source -> edit

affected -> send to LLM with image I_A:

"Rewrite BOTH source and edit with parallel sentence structure
and minimal word differences. Source must match the image;
edit must reflect the edit instruction. Same #sentences, same
#clauses, same rhythm -- swap only the minimum words that must
differ."
```

Figure 12: **Structural text alignment pipeline for continuous blending.** Stage 1 prompts the LLM to jointly generate aligned source/edit JSONs from two images while enforcing shared structure, object count, field order, and consistent wording for unaffected regions. Stage 2 performs image-aware bidirectional refinement: unchanged fields are force-copied, while affected fields are rewritten in parallel with minimal word substitutions to preserve structural alignment. This concentrates textual differences only on semantically modified content.

A.4 Additional Implementation Details

Structural Prompt Alignment. Figure 12 illustrates the two-stage LLM alignment pipeline used for continuous blending. We first jointly generate aligned source/edit JSON descriptions from the two input images while enforcing shared structure, field ordering, and consistent wording for unaffected content. We then perform image-aware refinement to localize textual differences only to semantically modified regions, while unchanged fields are force-copied to preserve exact correspondence. This stage produces consistent text-level alignment.

Runtime Overhead. The main preprocessing overhead comes from two Gemini calls for structural prompt generation and refinement, requiring 23.5 s per prompt pair on average on PIE-Bench. The two FIBO-edit (SmolLM) text-encoder passes together with the embedding alignment require approximately 84 ms, of which the alignment itself accounts for about 20 ms. These operations are performed only once per prompt pair and the resulting aligned embeddings are reused across all interpolation points, amortizing the preprocessing cost over the entire trajectory.

B Additional results

We include in the following figures additional qualitative results complementing the main paper for the two applications evaluated in this work: continuous editing and continuous blending. These supplementary results include comparisons and transition sequences generated using both FIBO-edit and FLUX2-Klein across diverse semantic transformations.

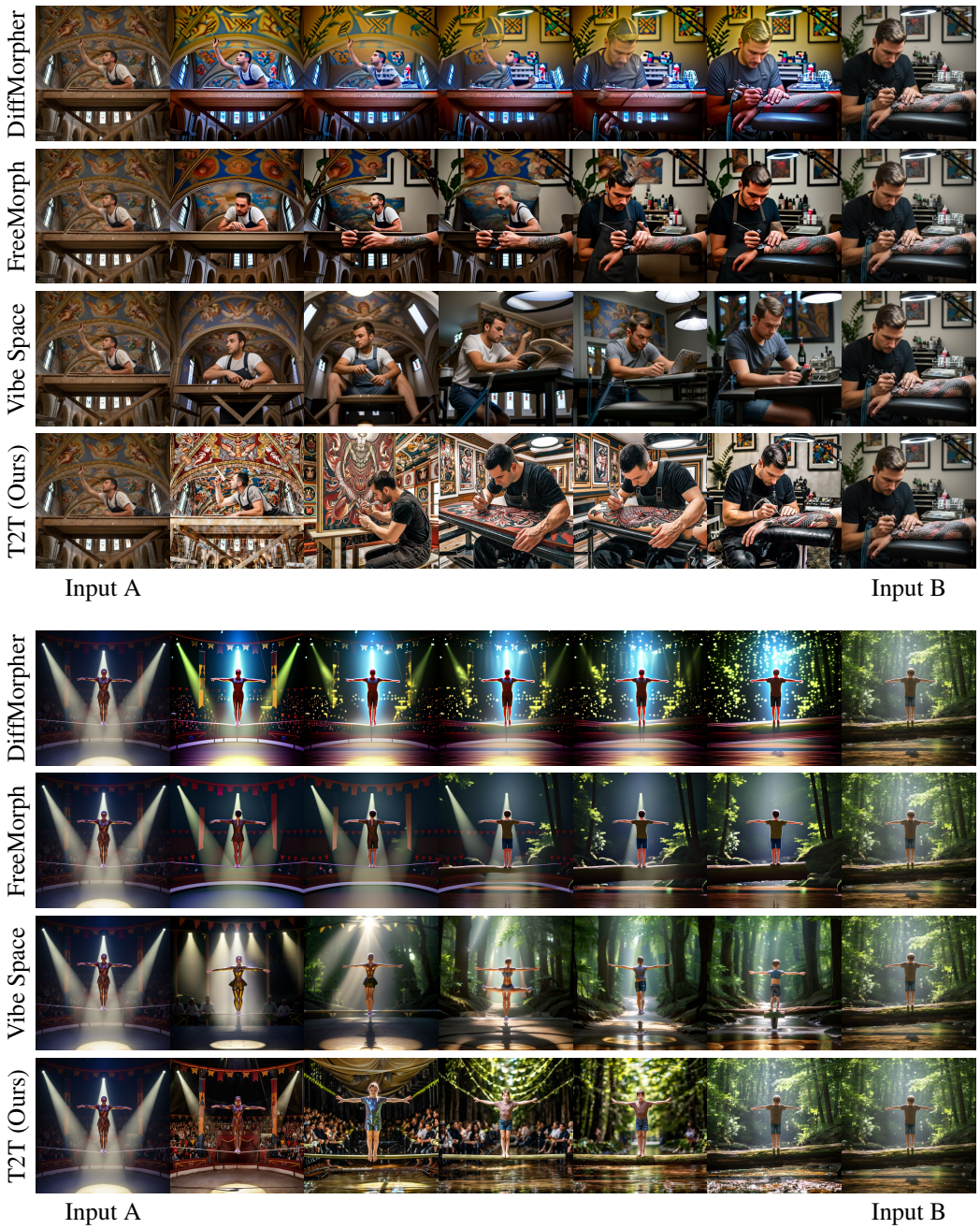


Figure 13: Qualitative comparison with continuous blending methods. Results generated using FIBO-edit.

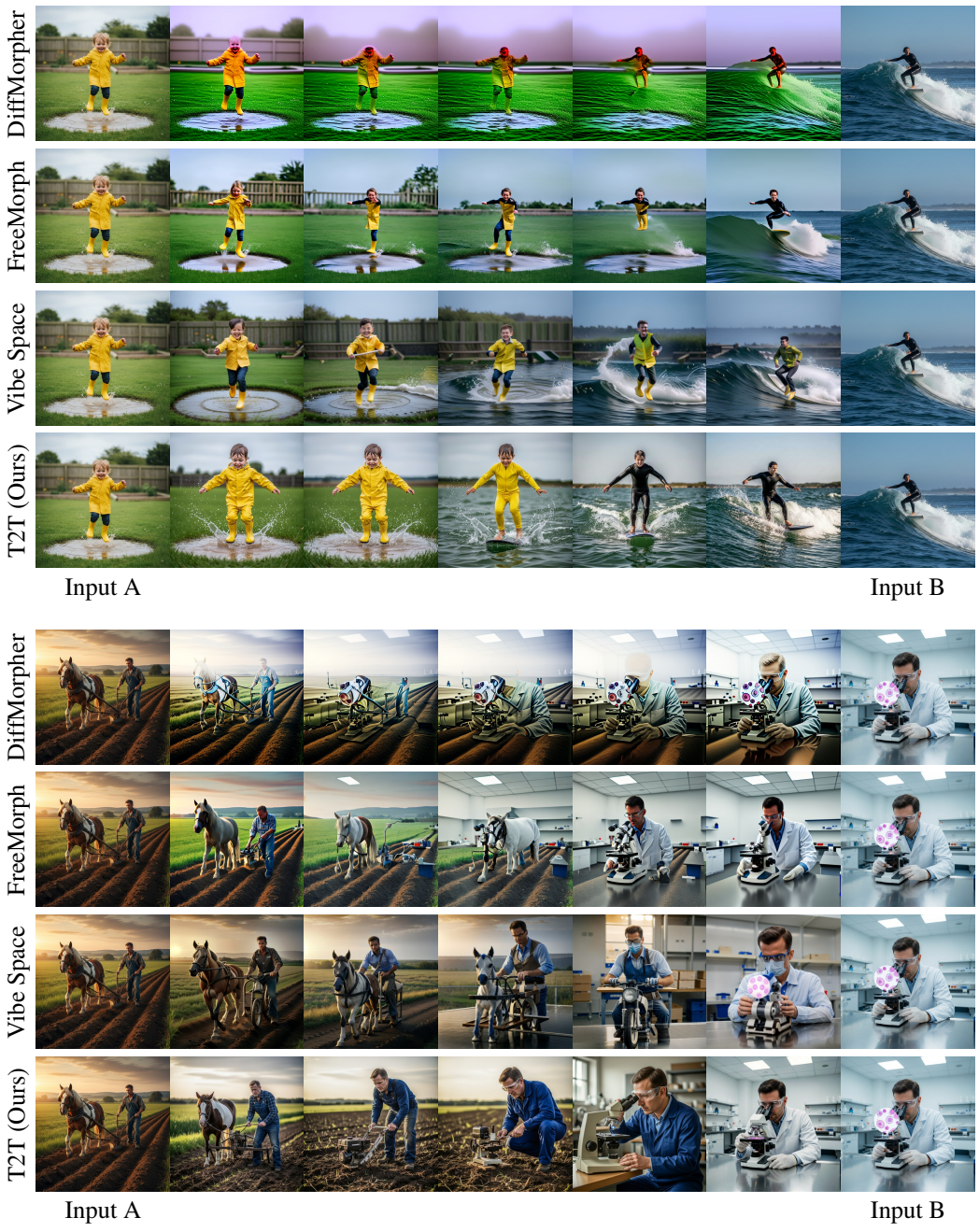


Figure 14: Qualitative comparison with continuous blending methods. Results generated using FLUX2-Klein.



Figure 15: Additional continuous blending results. Each row shows a gradual transition between two input images. Top row: results generated using FIBO-edit. Bottom two rows: results generated using FLUX2-Klein.

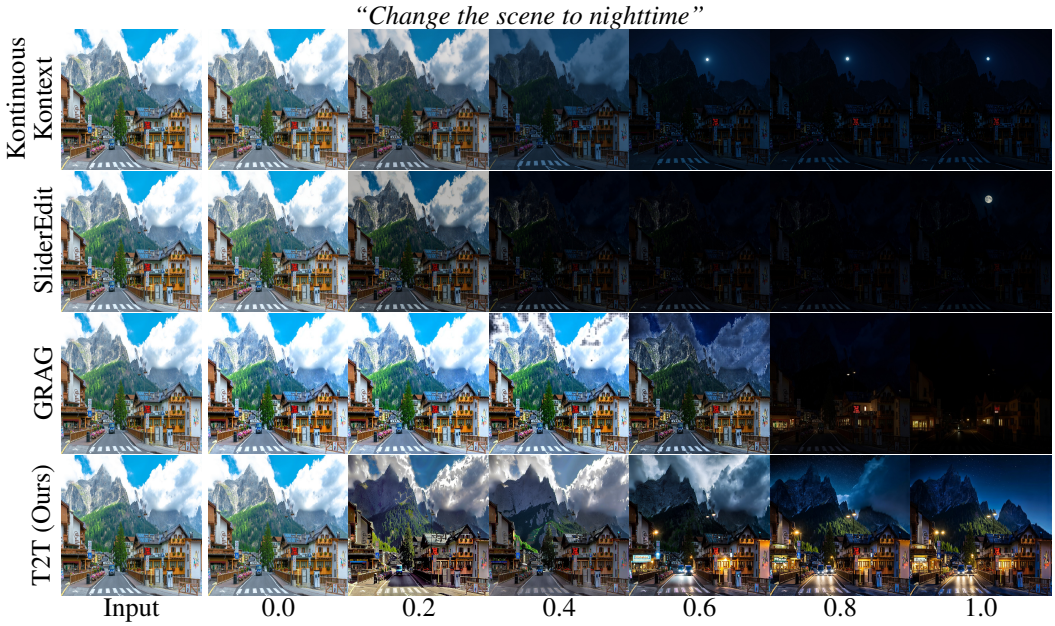


Figure 16: **Qualitative comparison with continuous editing methods**. For the instruction "change the scene to nighttime," prior approaches mainly rely on appearance-based transitions such as progressive darkening, whereas our method produces a more coherent semantic transition with gradual changes in illumination and shadow placement consistent with the changing time of day.

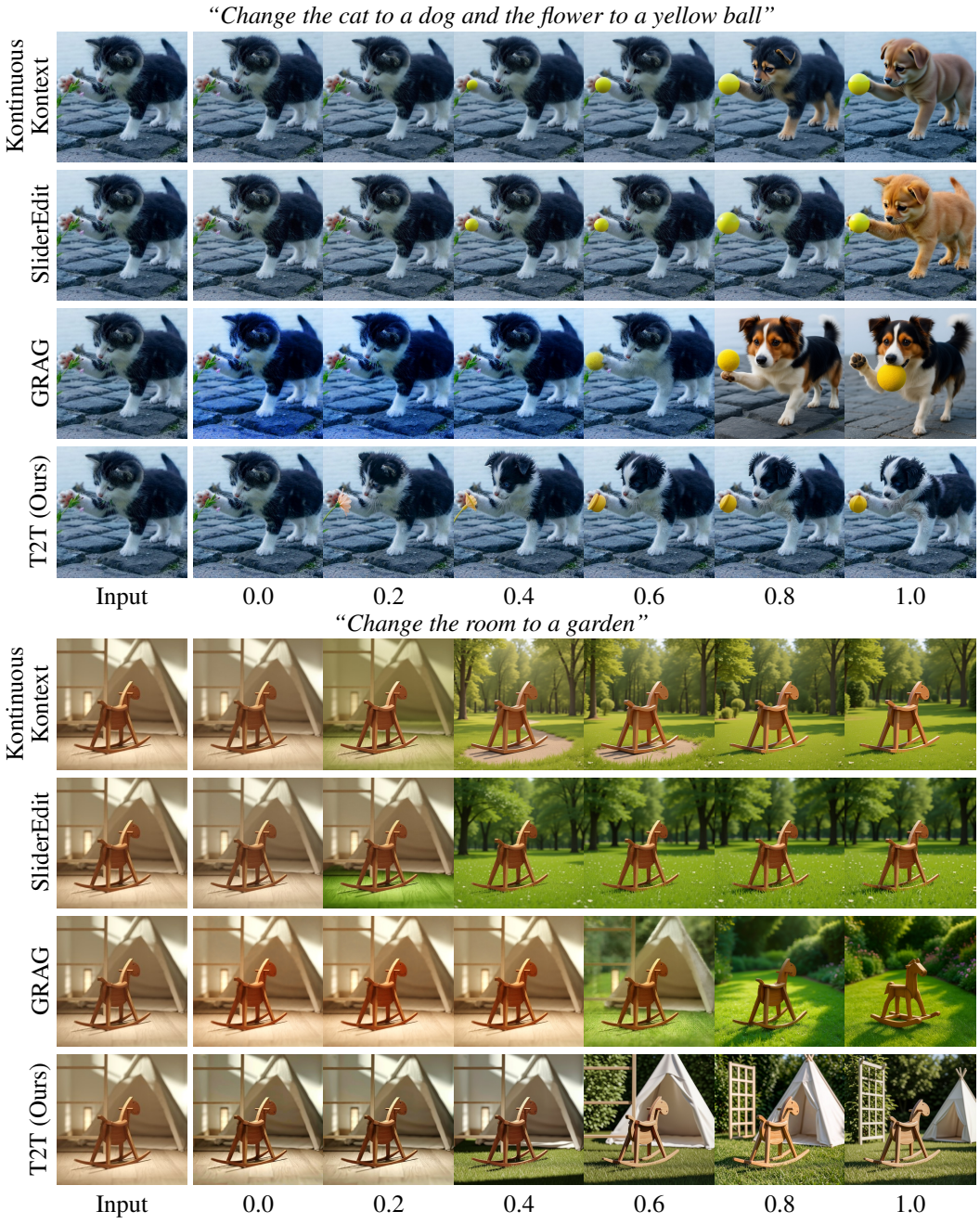
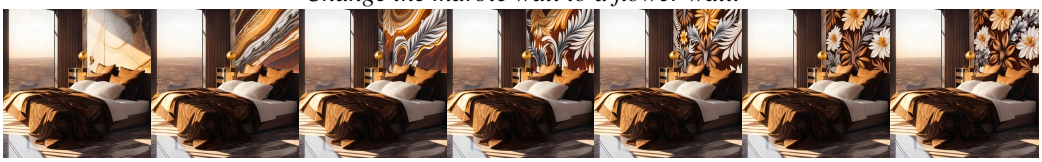


Figure 17: Qualitative comparison with continuous editing methods. Top example: results generated using FIBO-edit. Bottom example: results generated using FLUX2-Klein.

"Transform the wooden birdhouse into a white security camera."



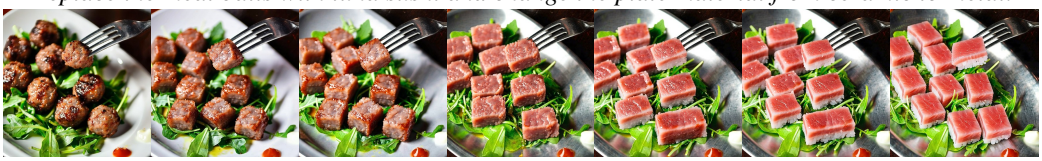
"Change the marble wall to a flower wall."



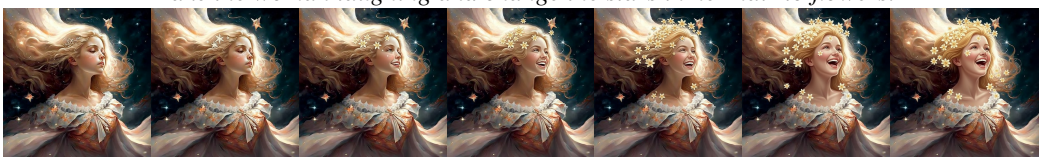
"Make the bacon strips crisp and burnt."



"Replace the meat balls with tuna sushi and change the plate material from ceramic to metal."



"Make the woman laughing and change the stars in her hair to flowers."



"Change the horse to a dog and its action from running to jumping."



"Change the rose from blooming to dried."



Input Edit Intensity $\longrightarrow$

Figure 18: Additional continuous editing results. Each row shows a gradual edit applied to a reference image along the specified direction. Top three rows: results generated using FIBO-edit. Bottom four rows: results generated using FLUX2-Klein.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contribution of aligning embedding representations for continuous blending, and these claims are supported by the proposed method and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the conclusion, including reliance on alignment quality and challenges in handling complex or non-linear semantic transformations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides sufficient details on the method, datasets, models, and key parameters to reproduce the main experimental results, with additional details provided in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code is not publicly available at submission time, and the dataset will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the experimental setup, including datasets, models, and key parameters, with additional details provided in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report error bars or statistical significance, as the evaluation focuses on standard metrics and user study comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper does not report detailed compute resources, as experiments are conducted at inference time using standard pre-trained models without additional training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not discuss broader societal impacts, as it focuses on a methodological contribution, although the method relates to generative image modeling and may have implications such as image manipulation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release models or datasets at submission time, and therefore safeguards for controlled release are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, models, and prior methods used in the paper are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper introduces a new dataset, but it is not released or documented at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The paper includes a user study, and participant instructions and details of the study setup, including the absence of compensation, are provided in the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The paper includes a user study but does not report IRB approval or formal risk assessment procedures.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper uses an LLM as part of the method for text-level alignment, and this usage is described in the experimental setup.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.