CamEdit: Continuous Camera Parameter Control for Photorealistic Image Editing

Xinran Qin[1]*, **Zhixin Wang**[2]*, **Fan Li**[2], **HaoYu Chen**[3],
RenJing Pei[2], **WenBo Li**[4], **XiaoChun Cao**[1]†
[1]Shenzhen Campus of Sun Yat-sen University [2]Huawei Noah's Ark Lab
[3]HKUST (Guangzhou) [4]CUHK

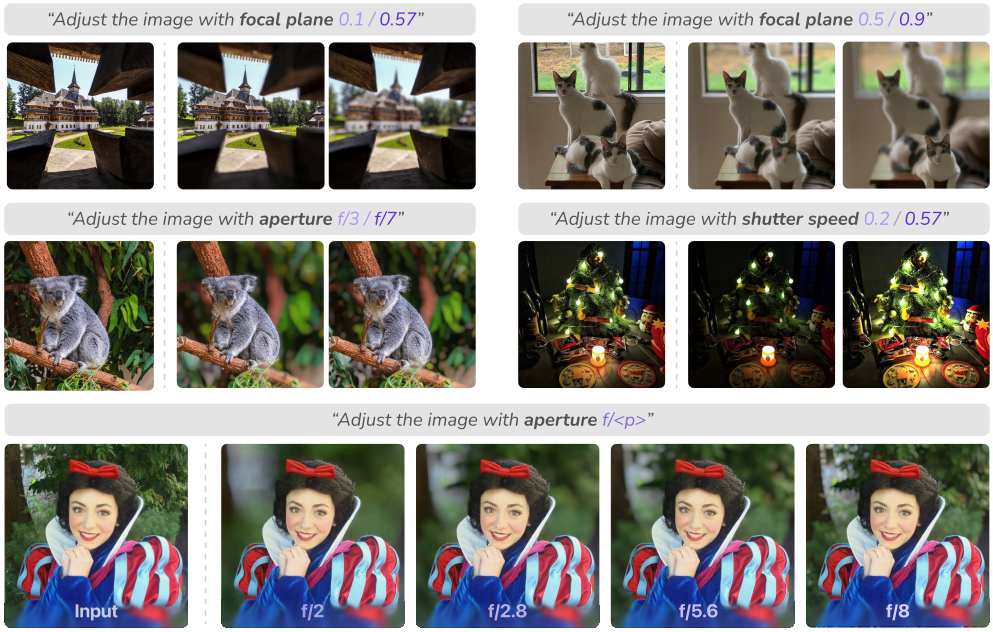

Figure 1: The proposed CamEdit enables photorealistic image editing through manual textual input of continuous camera parameters, including aperture, focal plane, and shutter speed, resulting in visually realistic outcomes.

Abstract

Recent advances in diffusion models have substantially improved text-driven image editing. However, existing frameworks based on discrete textual tokens struggle to support continuous control over camera parameters and smooth transitions in visual effects. These limitations hinder their applications to realistic, camera-aware, and fine-grained editing tasks. In this paper, we present CamEdit, a diffusion-based framework for photorealistic image editing that enables continuous and semantically meaningful manipulation of common camera parameters such as aperture and shutter speed. CamEdit incorporates a continuous parameter prompting mechanism and a parameter-aware modulation module that guides the model in smoothly adjusting focal plane, aperture, and shutter speed, reflecting the effects of varying camera settings within the diffusion process. To support supervised learning in this setting, we introduce CamEdit50K, a dataset specifically designed for photorealistic image editing with continuous camera parameter settings. It contains over 50k image pairs combining real and synthetic data with dense camera

∗ Equal Contribution † Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

parameter variations across diverse scenes. Extensive experiments demonstrate that CamEdit enables flexible, consistent, and high-fidelity image editing, achieving state-of-the-art performance in camera-aware visual manipulation and fine-grained photographic control.

1 Introduction

Recently, diffusion models [1, 2, 20, 49, 50, 52, 54, 51, 32] have become powerful tools for both image generation and editing. They usually apply a pre-trained text encoder such as CLIP [48] and T5 [66] to inject manual textual prompts information into the generation process, enabling better generation quality and more precise control. Meanwhile, as social media platforms grow and smartphone cameras continue to advance, editing images to reflect photorealistic optical effects has become practically valuable. This highlights the need for editing methods that can directly manipulate camera parameters. However, most existing image editing methods [4, 5, 19, 23, 24, 29, 35, 58, 62] focus mainly on three main tasks: semantic editing, stylistic editing and structural editing.

Few prior works target photorealistic image editing, which edits indistinguishable from real photographs through precise control of camera parameters. In this work, we focus on the precise adjustments of focal plane[1] , aperture and shutter speed in camera parameters, which play a fundamental role respectively in determining focal range, background defocus degree, and exposure time [46, 57] during the photo-taking process.

Diffusion models capture strong spatial priors and scene geometry [55], making them suited for photorealistic editing. Recent works encode camera settings as discrete tokens within text-to-image (T2I) or text-to-video (T2V) generation frameworks [11, 70]. However, such discrete textual token-based approaches are difficult to directly apply to editing tasks involving continuous camera parameter control through textual prompt input (e.g. "*Adjust the image with aperture* $f/2.8$", etc.). This mismatch hampers smooth parameter adjustment and limits applicability to photographic editing.

To overcome these challenges, we introduce **CamEdit**, a diffusion-based framework for photorealistic image editing that allows continuous control of camera settings using text prompts. Instead of turning parameter values into separate tokens, we propose a continuous parameter prompting method, which interpolates between predefined anchor embeddings in the text space. This preserves alignment with representation distribution of the pre-trained model while enabling fine-grained control over a wide range of settings (such as "aperture $f/[2, 10]$", "shutter speed $[0, 1]$"). As diffusion backbones lack explicit camera priors and fail to capture parameter-specific effects, we further propose a *parameter-aware modulation* module that conditions spatial and channel features throughout the diffusion transformer, making explicit both local and global effects that text embeddings alone miss.

Given the lack of high-quality datasets for photorealistic camera-aware editing, we construct a hybrid dataset named **CamEdit50K**, which includes real-world photographs with extracted or estimated EXIF metadata[2], along with synthetic image pairs rendered under controlled variations in focal plane, aperture, and shutter speed. This dataset provides a strong foundation for learning models that are physically consistent and aware of the effect of varying camera parameters.

In summary, our main contributions can be summarized as follows:

- We propose **CamEdit**, a diffusion-based framework for photorealistic image editing that enables continuous and fine-grained control over intrinsic camera parameters such as aperture, focal plane, and shutter speed, entirely through manual textual prompts.
- We design a *continuous parameter prompting* mechanism and a *parameter-aware modulation* module to enable smooth and physically consistent control across varying camera settings.
- We construct a dataset, **CamEdit50K**, which contains aligned image pairs and corresponding camera parameter instructions, addressing the lack of supervised data for photorealistic image editing with continuous camera parameter.

[1]Focal plane is corresponding to the focal point in camera settings.

[2]EXIF is metadata embedded in image files that records camera settings such as aperture and shutter speed.

2 Related Work

Image Editing with Diffusion Models. Recent diffusion-based generative methods such as Imagen [54] and DALLE [49] leverage diffusion models conditioned on manual textual prompts to control the generation process. Consequently, Textual Inversion [14] and DreamBooth [53] allow for personalized image generation base on diffusion models. ControlNet [71] adds more control by using conditions like depth, edges, or pose. LoRA-based techniques [22, 34, 35] update only a small part of the model for quick adaptation. Unlike image generation, image editing modifies the style, structure, or content of an existing image to achieve specific goals. Prompt-to-Prompt [19] manipulates cross-attention between source and target prompts to guide the editing process, while InstructPix2Pix (IP2P) [5] extends this paradigm by fine-tuning diffusion models on synthetic triplets of (image, instruction, target). More recent approaches, such as InstructDiffusion [17] and MGIE [13], unify instruction-driven editing across a broad range of tasks and datasets, advancing general-purpose visual manipulation. To improve spatial fidelity, follow-up works incorporate localization priors such as masks and bounding boxes [58, 24, 4] to better preserve background consistency. Other efforts explore multi-task instruction tuning on large-scale synthetic datasets [37, 69, 64], enabling finer-grained semantic control. Approaches with sliders [16, 15] enable continuous control over the attributes of image edits. Despite these advances, existing frameworks remain centered on semantic and stylistic edits, which are rarely considering the growing needs for photorealistic editing.

Camera-Aware Models. Traditional camera-aware editing methods have shown strong performance in tasks such as focal plane adjustment [57, 45, 46, 63] and aperture simulation [7, 56]. These methods are typically based on physically inspired image formation models and often require additional inputs such as depth maps or aperture geometry [57, 45, 46, 18, 61, 63]. However, their applicability is generally limited to single-purpose editing scenarios due to their dependence on auxiliary data and restrictive modeling assumptions.

Recent diffusion-based approaches introduce camera control into generative pipelines, mainly focusing on extrinsic parameters like pose, viewing angle [8, 21, 33], or motion trajectory in text-to-video generation [40, 65, 68]. Conditioning is commonly achieved via camera tokens or scene descriptions to enable view synthesis and motion control. Some methods embed camera parameters as discrete tokens into T2I [11, 12] and T2V [70] diffusion models to generate images with varying physical properties. However, these approaches face two main limitations: they do not support editing real images in a physically consistent manner, and they represent continuous camera parameters in a discretized form, which restricts control precision and limits generalization.

3 CamEdit50K Dataset

Existing camera-aware datasets [11, 56, 70, 7] mainly focus on generation tasks, often lacking aligned image pairs or sufficient variation in camera parameters and content diversity. To address these limitations, we introduce CamEdit50K, a dataset specifically designed for photorealistic image editing under continuous, physically grounded camera control.

As shown in Table 1 and Figure 2, CamEdit50K unifies paired real and synthetic imagery, multi-parameter coverage, and explicit camera settings to support camera-aware editing and evaluation. Real photos supply rich content but often lack complete metadata. We recover missing parameters through a real-data parameter estimation pipeline. Synthetic images are produced with a synth-data rendering pipeline and come with ground-truth camera values, which enables accurate and dense supervision. By integrating these complementary sources, CamEdit50K delivers diversity and controllability, enabling continuous camera-parameter editing.

Real-Data Param Estimation. For the majority without metadata, we estimate these parameters using physically grounded methods: (i) *Focal Plane:* To ensure consistency across images, we define the focal plane within a normalized depth range $[0, 1]$, representing focus from far to near. Depth maps are predicted by `Depth Anything V2` [67] and normalized accordingly. The in-focus region is identified by comparing the target image with an all-in-focus input [60], and its mean depth is used as the estimated focal plane. (ii) *Aperture:* Since aperture primarily governs background defocus, we measure blur using an edge-based estimation [27]. The estimated defocus level is then converted to an effective aperture diameter via a simplified thin-lens model [44]. (iii) *Shutter Speed:* We estimate

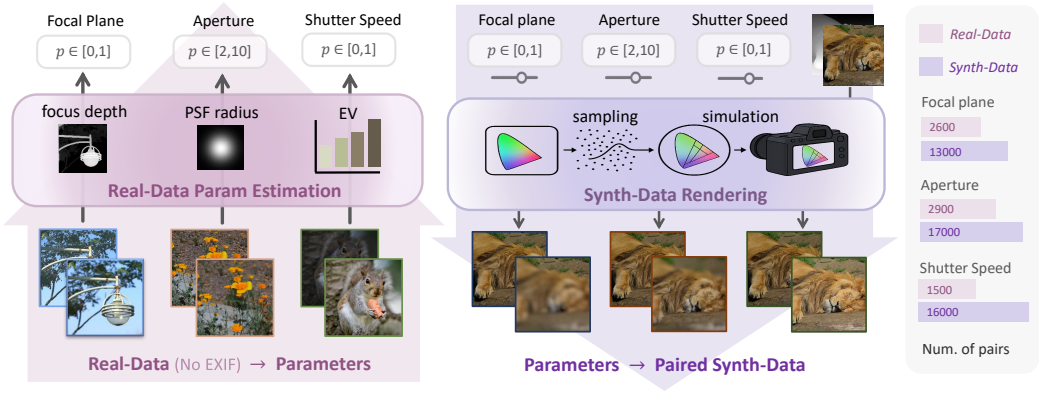

Figure 2: Overview of the CamEdit50K construction pipeline across focal plane, aperture, and shutter speed. Real image pairs without EXIF metadata are parameterized using physical cues such as focus depth, PSF radius, and exposure value. Synthetic pairs are generated by sampling camera settings and rendering photorealistic images under controlled conditions. The chart on the right shows the number of images for each parameter in CamEdit50K.

Table 1: Comparison with existing camera-aware datasets.

Dataset	Venue	#Samples	Task	Real-Data	Synth-Data	Synth-Realism	Parameter-Dense	Scene Diversity
RealBokeh [56]	Arxiv 2025	23k	Render	✓	✗	-	✗	✓
Camera20k [11]	SA 2025	20k	Generate	✓	✗	-	✗	✗
PhotoGen [70]	CVPR 2025	3k[π,3]	Generate	✗	✓	✗	✓	✗
CamEdit50K	–	54k	**Edit**	✓	✓	✓	✓	✓

exposure time from global brightness statistics and invert the camera response using a differentiable ISP pipeline [9, 31, 36], yielding a shutter speed consistent with the observed luminance.

Synth-Data Rendering. We generate synthetic image pairs by sampling camera parameters and rendering the corresponding effects. (i) *Focal Plane:* We sample the focal plane from the interval $[0, 1]$, selecting depths ranging from background to foreground. To simulate realistic depth-of-field effects, we employ a differentiable bokeh renderer [57]. (ii) *Aperture:* Using the depth map from [67], we fix the focal plane on the foreground and apply `BRIA.AI` [3] matting to preserve sharpness in the focused region. A thin-lens renderer [45] is then used to simulate varying aperture from $f/2$ to $f/10$, producing different degrees of defocus blur. (iii) *Shutter Speed:* We simulate exposure durations within the range $[0, 1]$ seconds by adjusting radiance in the HDR domain and converting it to RGB through a differentiable ISP pipeline [9, 31, 36].

4 Method

Our CamEdit framework adopts the instruction-driven editing paradigm of IP2P [5], while building upon the diffusion backbone as illustrated in Figure 3. Given a camera-parameter instruction, we first apply continuous parameter prompting as described in Section 4.1, which enables fine-grained prompt conditioning. The resulting embeddings, combined with the input image, are then fed into the transformer equipped with parameter-aware modulation modules described in Section 4.2, which inject parameter-specific feature modulation into the generation process.

4.1 Continuous Parameter Prompting

Directly learning parameter embeddings and appending them to other text features, while bypassing the text encoder, introduces a distributional mismatch with the frozen text embedding space. This misalignment degrades generation quality and hampers convergence, as shown in Section 5.3. To address this, our continuous parameter prompting synthesizes parameter representations by interpolating between anchor embeddings of adjacent discrete tokens within the text embedding space. The

[3]PhotoGen synthesizes samples by continuously sampling camera parameters over 3K fixed images, reducing scene diversity and affecting realism.

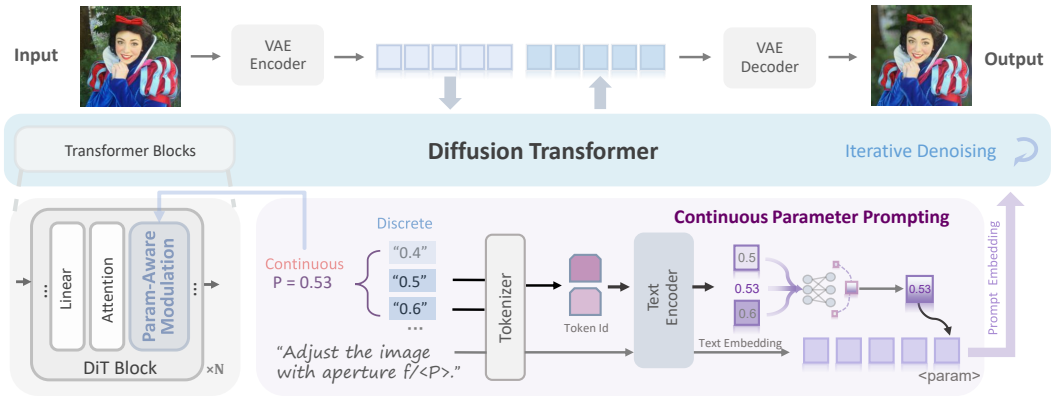

Figure 3: Framework Overview. The continuous parameter prompting module obtains a continuous parameter embedding via learnable interpolation over discrete camera token embeddings. This embedding replaces the placeholder token in the text embedding space, while preserving the original prompt structure. The parameter is also injected into the diffusion transformer via the parameter-aware modulation module, which adjusts features to reflect the corresponding visual effects.

resulting embedding replaces the parameter placeholder in the encoded prompt, ensuring perceptual continuity across parameter variations. This mechanism requires no modification to the text encoder and integrates seamlessly into diffusion-based frameworks.

Let $p \in \mathbb{R}$ denote a continuous camera parameter, and $\{p_1, \ldots, p_K\}$ be a set of predefined discrete anchor values with associated learnable embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_K\} \subset \mathbb{R}^d$, where each $\mathbf{e}_k$ is obtained by encoding the anchor token via the frozen CLIP tokenizer and text encoder. For any $p \in [p_i, p_{i+1}]$, the parameter embedding is computed as:

$$\mathbf{e}_p = \text{Linear}([\mathbf{e}_i, \mathbf{e}_{i+1}]) + \text{MLP}(\phi(p)), \tag{1}$$

where $\phi(p) \in [0, 1]$ represents the normalized relative position of p between anchors p_i and p_{i+1}. The linear projection aggregates the semantic content of the two neighboring embeddings, while the MLP, implemented as a two-layer feed-forward network with ReLU activation, introduces a position-dependent residual to capture fine-grained variation. The final embedding $\mathbf{e}_p$ replaces the parameter placeholder in the encoded text prompt representation.

4.2 Parameter-Aware Modulation

Camera parameter variations primarily influence the spatial appearance of an image while preserving its underlying semantic content. Such changes include spatial transformations, for example, depth-dependent focus shifts across foreground and background regions [59]. Additionally, parameters such as shutter speed induce global exposure changes [30], motivating channel-wise feature modulation.

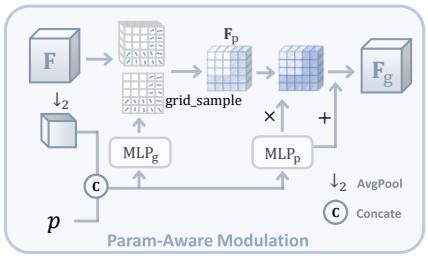

Figure 4: Illustration of parameter-aware modulation.

To effectively model visual effects, we modulate intermediate features conditioned on the parameter p through two complementary operations as shown in Figure 4. To improve parameter sensitivity and information flow, the modulation is applied after the self-attention layers in each transformer block, where features contain rich contextual dependencies. The first component, geometry-aware spatial modulation, models lens-induced geometric distortion and depth-dependent focus transitions. It predicts a spatial displacement field that perturbs feature coordinates based on visual features and the input parameter. Specifically, we apply 2×2 average pooling to the input feature map $\mathbf{F}$ and feed the pooled representation, together with p, into a parameter-adaptive MLP:

$$\Delta\mathbf{G} = \text{MLP}_\text{g}(\texttt{AvgPool}(\mathbf{F}), p) \in \mathbb{R}^{2\times H\times W}, \tag{2}$$

where $\texttt{AvgPool}(\cdot)$ denotes average pooling over non-overlapping 2×2 patches. The warped feature is then computed via:

$$\mathbf{F}_{\text{g}} = \texttt{grid\_sample}(\mathbf{F}, \mathbf{G}_{\text{base}} + \Delta\mathbf{G}), \tag{3}$$

where $\mathbf{G}_{\text{base}}$ represents the canonical coordinate grid of $\mathbf{F}$ as defined in [25] and $\texttt{grid\_sample}$ performs differentiable sampling of $\mathbf{F}$ at continuous coordinates.

The second component, channel-wise modulation, adjusts feature amplitudes to capture global appearance variations. We compute channel-wise scaling and bias terms as follows:

$$\mathbf{F}_{\text{p}} = \gamma(p) \cdot \mathbf{F}_{\text{g}} + \beta(p), \quad \text{where } \gamma(p), \beta(p) = \text{MLP}_{\text{p}}(\texttt{AvgPool}(\mathbf{F}), p) \in \mathbb{R}^C. \tag{4}$$

Both MLP_{g} and MLP_{p} are implemented as single-layer feed-forward networks with intermediate ReLU activation. Then $\mathbf{F}_{\text{p}}$ is forwarded to the subsequent transformer block.

5 Experiments

5.1 Implementation Details

Training and Inference. We adopt Stable Diffusion 3 (SD3) [10] as our backbone due to its strong generation quality and color fidelity. The majority of SD3 weights are kept frozen to preserve its pre-trained capacity. We update only the text embedding layer to learn anchor tokens and enable our learnable parameter interpolation. For each task, we predefine 10 anchor tokens via the tokenizer to guide training. The transformer is initialized from [72], and we fine-tune lightweight adapters using LoRA [22], combined with our physics-driven adaptation module. We train the model using AdamW [38], with learning rates of 1e-5. Training is conducted on 512×512 resolution images with a batch size of 32 for 50 epochs. During inference, the model requires only an input image and an instruction specifying any continuous camera parameter value within the valid range.

Metrics. We evaluate performance from three perspectives: perceptual quality, content preservation, and parameter control accuracy. Perceptual quality is measured using NIQE [42] and MUSIQ [28], which assess naturalness and visual fidelity without references. Content preservation is quantified via DINO similarity [6], capturing semantic alignment between the source and edited images. To assess parameter control, we compute the L1 error between the instruction-specified target and the estimated parameter extracted from the generated image. Estimation follows the physically grounded procedure described in Section 3. We evaluate 200 images across varying parameter settings.

5.2 Comparison to State-of-the-Art Methods

Comparison Methods. We firstly compare our method against state-of-the-art diffusion-based baselines, including editing models such as SuperEdit [41], UltraEdit [72], and In-Context Edit [47], as well as the camera-aware I2V model PhotoGen [70]. To evaluate PhotoGen [70], we generate images using prompts sampled from GPT-4o to simulate realistic text-based generation requests. We retrain UltraEdit on our CamEdit50K to enable camera-aware editing, denoted as UltraEdit*.

Beyond diffusion-based baselines, we further compare our method with other approaches across all editing tasks. For aperture editing, we compare with BokehMe [45], BRVIT [43], and DrBokeh [57]. For focal plane editing, we evaluate against BokehMe [45], MPIB [46], and DrBokeh [57]. For shutter speed editing, we include advanced low-light and exposure-aware methods such as SCI [39], CycleR2R [36], and CLODE [26]. All methods are tested under a consistent exposure configuration and evaluated for their ability to adapt image brightness while preserving both structural integrity and perceptual quality.

Quantitative Comparison. As shown in Table 2 (a), our method consistently outperforms other editing across all tasks in NIQE, DINO, and control error. Compared to the retrained UltraEdit* model, our method achieves an over 40% relative reduction in average control error, demonstrating substantially improved parameter alignment. Existing editing models do not explicitly incorporate camera parameters, limiting their ability to generate parameter-consistent results. Although UltraEdit* benefits from CamEdit50K supervision, it remains less precise than our approach.

We further evaluate performance via GPT-4o across photographic realism, content preservation, and parameter accuracy. We further assess performance using GPT-4o evaluation and a user study with 15 photography experts. Each participant evaluated 20 image sets per task across the three tasks, scoring

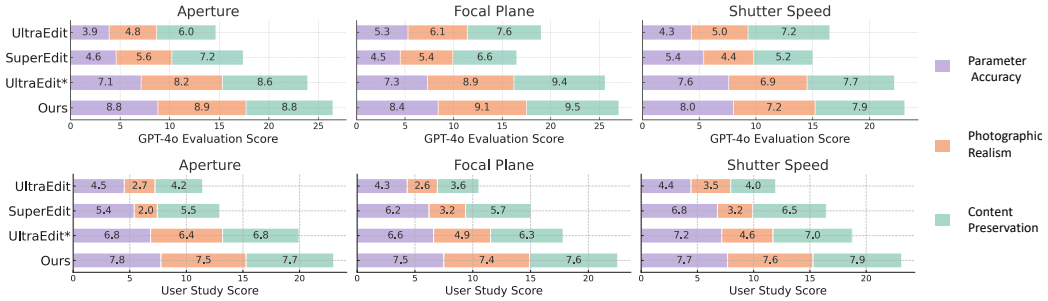

Figure 5: GPT-4o evaluation and user study across three dimensions: photographic realism, content preservation, and camera parameter accuracy, using a 0–10 scale (higher is better).

Table 2: Quantitative comparison across camera-aware editing methods under aperture, focal-plane, and shutter-speed control. Best results are in **bold**.

(a) Diffusion-Based Methods

Method	Aperture			Focal Plane			Shutter Speed		
	NIQE↓	DINO↑	Error↓	NIQE↓	DINO↑	Error↓	NIQE↓	DINO↑	Error↓
SuperEdit [41]	4.43	0.73	∼5	5.28	0.73	∼0.5	4.86	0.75	∼0.5
UltraEdit [72]	4.58	0.70	∼5	5.57	0.74	∼0.5	5.02	0.72	∼0.5
In-Context Edit [47]	4.30	0.78	∼5	4.91	0.80	∼0.5	4.29	0.81	∼0.5
PhotoGen [70]	6.65	–	1.31	6.21	–	0.28	6.11	–	0.19
UltraEdit* [72]	4.21	0.78	1.87	5.14	0.79	0.25	4.69	0.88	0.23
Ours	**3.34**	**0.83**	**0.60**	**4.46**	**0.82**	**0.15**	**4.28**	**0.93**	**0.11**

(b) Other Methods

Method	Aperture				Method	Focal Plane			
	NIQE↓	MUSIQ↑	DINO↑	Error↓		NIQE↓	MUSIQ↑	DINO↑	Error↓
BokehMe [45]	4.62	59.70	0.82	0.62	BokehMe [45]	5.19	48.97	0.79	0.22
BRVIT [43]	6.53	50.72	0.71	–	MPIB [46]	5.14	49.05	0.78	0.18
DrBokeh [57]	3.81	62.26	0.81	**0.58**	DrBokeh [57]	5.07	47.59	0.80	0.17
Ours	**3.34**	**62.91**	**0.83**	0.60	**Ours**	**4.46**	**52.64**	**0.82**	**0.15**

photographic realism, content preservation, and parameter accuracy. As shown in Figure 5, our method achieves the highest average scores on all dimensions.Relative to UltraEdit*, our CamEdit improves realism by 8% and parameter-control accuracy by 10%, demonstrating consistently higher realism, stronger content preservation, and more precise control.

As shown in Table 2 (b), our method consistently outperforms rendering-based baselines on most metrics, such as DrBokeh [57], BokehMe [45] on both aperture and focal plane editing tasks. For aperture editing, our approach yields 23% lower control error compared to baselines. Relative to the strongest competitor, DrBokeh, our method improves NIQE by 12% and reduces error by 24%. These results highlight the benefit of continuous parameter control in diffusion models, enabling physically grounded and perceptually faithful image editing.

Qualitative Comparison. As shown in Figure 6, our method achieves fine-grained and continuous control across all camera parameters, demonstrating clear parameter awareness. Existing diffusion-based editing models lack such a capability due to the absence of camera supervision during training. UltraEdit*, retrained on our dataset, shows improvement, but its discrete prompting leads to occasional mismatches when interpolating unseen values. In contrast, our method ensures smooth transitions and better fidelity, particularly around depth-sensitive regions such as foreground boundaries, owing to our parameter-aware modulation. We also compare with the generative model PhotoGen [70], where our results exhibit higher photographic realism and more coherent spatial structure. This improvement stems from our editing-based formulation and the use of real image pairs during training. Our method handles diverse scenarios with realistic parameter effects, such as light flares from aperture adjustment in nighttime scenes or exposure refinement that enhances visual aesthetics.

Figure 7 further supports our findings. Under aperture variation, our method preserves fine details like hair strands and object contours. For focal plane editing, we maintain sharpness in in-focus areas,

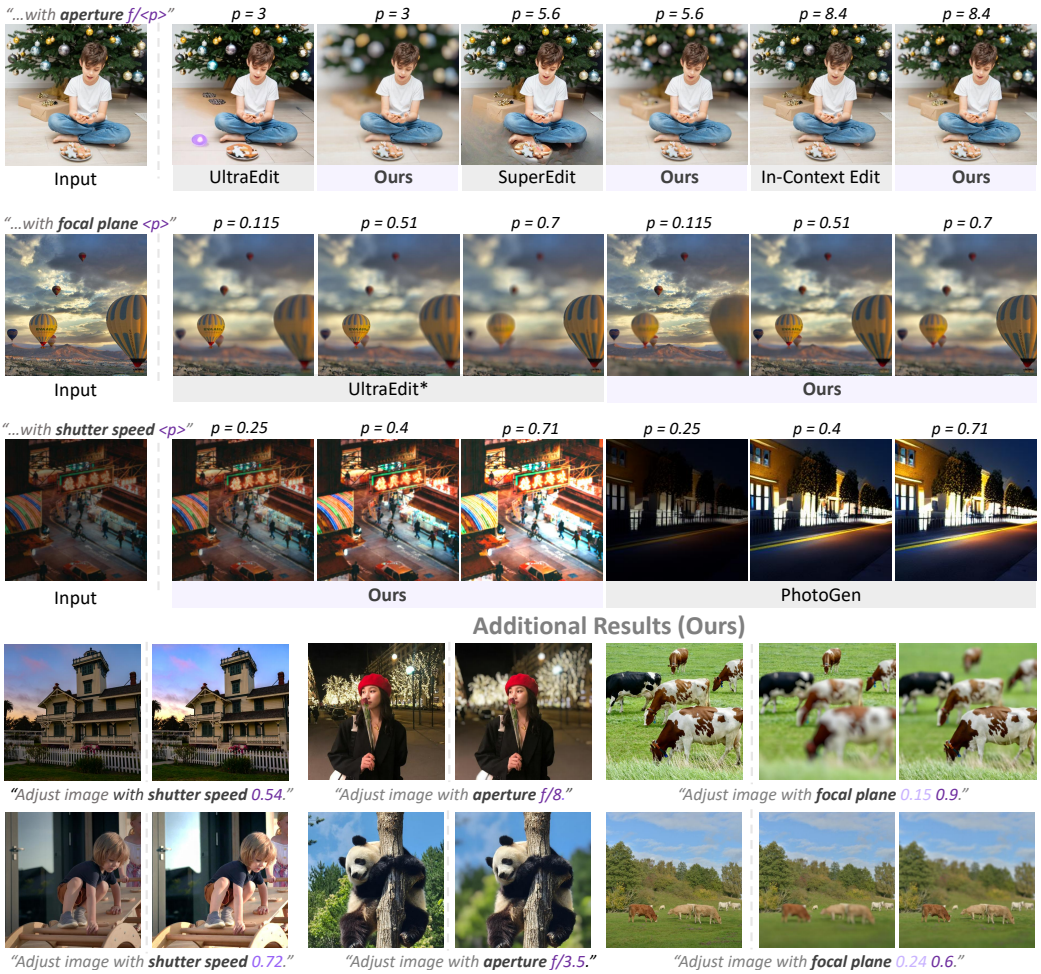

Figure 6: Visual comparison with diffusion-based methods. Our results illustrate smooth and perceptually consistent edits under various instructions.

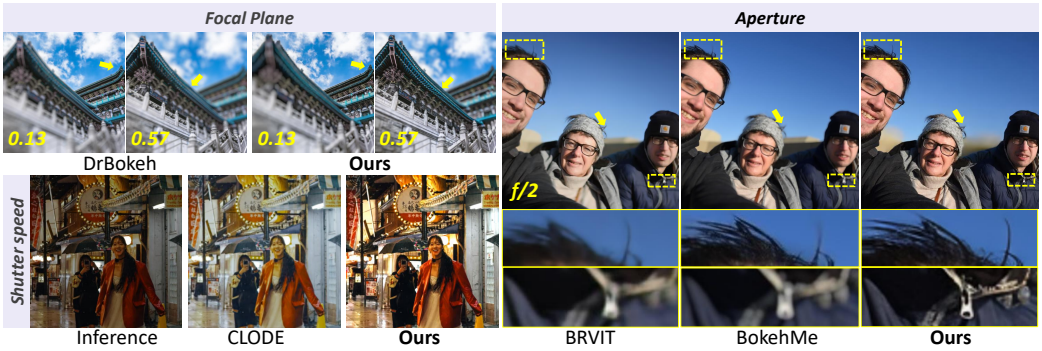

Figure 7: Visual comparison with other methods. Our method achieves finer detail and more realistic photographic effects compared to traditional rendering pipelines.

especially on architectural edges. In shutter speed control, our outputs adjust motion-related brightness while preserving photographic style. These results confirm that CamEdit delivers physically consistent edits with precise control and high visual fidelity.

5.3 Ablation Study

In this section, we analyze the components of CamEdit and the composition of CamEdit50K. All ablations are conducted on the focal-plane task.

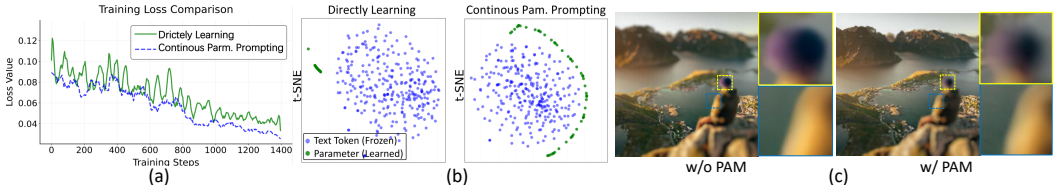

Figure 8: (a) Training loss comparison between direct parameter embedding and ConPrompt. (b) T-SNE visualization showing more semantically aligned embeddings with ConPrompt. (c) Visual comparison with and without PAM.

Table 3: Ablation study on key components of our CamEdit.

Method	NIQE↓	MUSIQ↑	DINO↑	Error↓
w/o ConPrompt	4.31	54.26	0.80	0.25
w/ Direct Embed.	5.25	52.41	0.81	0.27
w/o PAM	4.79	54.10	0.82	0.18
Ours (Full)	**4.46**	**52.64**	**0.83**	**0.15**

Table 4: Performance with different synthetic-to-real ratios of training dataset.

Syn:Real	NIQE↓	MUSIQ↑	DINO↑	Error↓
0 : 1	4.81	53.73	0.75	0.23
1 : 1	4.76	53.92	0.80	0.21
1 : 0	4.93	53.14	0.76	**0.15**
CamEdit50K	**4.46**	**52.64**	**0.83**	**0.15**

Continuous Parameter Prompting. As shown in Table 3, removing continuous parameter prompting (w/o ConPrompt), leads to degraded control accuracy and lower perceptual quality. Similar inconsistencies are observed in the retrained UltraEdit*, where editing results lack smooth transitions and do not align well with the target parameters, as illustrated in Figure 6.

We further evaluate a direct parameter embedding learning variant (w/ Direct Embed.), which bypasses the text encoder and learns parameter embeddings independently. It results in lower visual quality, and unstable training, as shown by higher loss and slower convergence in Figure 8 (a) and (b). In contrast, ConPrompt interpolates between pre-defined anchor tokens within the frozen text space, yielding smooth, semantically meaningful embeddings that improve control, fidelity, and stability.

Parameter-Aware Modulation. Parameter-aware modulation improves both parameter accuracy and image quality, as evidenced by the performance drop in the "w/o PAM" variant in Table 3. This is because different camera parameters induce global shifts in scene appearance, and PAM enables the model to adapt feature representations accordingly. As shown in Figure 8 (c), removing PAM results in unnatural transitions in defocus regions, particularly around human silhouettes, where the blur at object boundaries becomes abrupt. We also evaluate the injection of continuous camera-parameter features into the diffusion timestep embeddings in place of PAM. Compared with CamEdit, NIQE is higher by 0.30, DINO is lower by 0.03, and Error is higher by 0.09, indicating weaker parameter control due to the absence of localized spatial modeling.

Composition of CamEdit50K. We analyze CamEdit50K by varying the ratio of synthetic to real data, as shown in Table 4. Training on real data alone lacks scene diversity and parameter coverage, leading to weaker perceptual quality and control. Adding synthetic data at a 1:1 ratio improves over real-only training, though gains are limited by the smaller total size. Synthetic-only training scales well and enhances control, but visual fidelity lags without real-image guidance. Our CamEdit50K, combining available synthetic and real data, delivers the best parameter control and visual quality, driven by the scale of synthetic data and the fidelity of real data.

6 Conclusion

We present CamEdit, a framework for photorealistic image editing with continuous control over camera parameters such as aperture, focal plane, and shutter speed. It features a parameter-aware design and is supported by CamEdit50K, a hybrid dataset with paired images and varying camera settings. CamEdit enables visually consistent, photorealistic edits and lowers the barrier for users to manipulate camera parameters through images, with potential applications in education, simulation, and creative industries. While effective on key controls, it does not yet support all camera parameters and cannot recover focus from blurred regions, which remains fundamentally challenging and presents a valuable direction for future research.

6.1 Acknowledgments

This work was supported in part by the Shenzhen Science and Technology Program (No. KQTD20221101093559018), the National Natural Science Foundation of China (No. 62025604), and the CIE–Smartchip Research Fund (No. 2024-08). We gratefully acknowledge the creators and maintainers of the public datasets and open-source models used in this work.

References

[1] FLUX. `https://github.com/black-forest-labs/flux`.

[2] Stable Diffusion. `https://github.com/Stability-AI/StableDiffusion`.

[3] Bria-ai. *https://huggingface.co/briaai/RMBG-1.4*, 2024.

[4] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8861–8870, 2024.

[5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF Conference on International Conference on Computer Vision*, pages 9630–9640, 2021.

[7] Kang Chen, Shijun Yan, Aiwen Jiang, Han Li, and Zhifeng Wang. Variable aperture bokeh rendering via customized focal plane guidance. *arXiv preprint arXiv:2410.14400*, 2024.

[8] Ta-Ying Cheng, Matheus Gadelha, Thibault Groueix, Matthew Fisher, Radomir Mech, Andrew Markham, and Niki Trigoni. Learning Continuous 3D Words for Text-to-Image Generation. *arXiv preprint arXiv:2402.08654*, 2024.

[9] Paul Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Special Interest Group on GRAPHics and Interactive Techniques*, pages 369–378, 1997.

[10] Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, abs/2403.03206, 2024.

[11] I-Sheng Fang, Yue-Hua Han, and Jun-Cheng Chen. Camera Settings as Tokens: Modeling Photography on Latent Diffusion Models. In *Special Interest Group on GRAPHics and Interactive Techniques Asia*, SA '24, pages 1–11, New York, NY, USA, December 2024. Association for Computing Machinery.

[12] Armando Fortes, Tianyi Wei, Shangchen Zhou, and Xingang Pan. Bokeh diffusion: Defocus blur control in text-to-image diffusion models. *arXiv preprint arXiv:2503.08434*, 2025.

[13] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[14] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

[15] 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.

[16] Rohit Gandikota, Zongze Wu, Richard Zhang, David Bau, Eli Shechtman, and Nick Kolkin. Sliderspace: Decomposing the visual capabilities of diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025. arXiv:2502.01639.

[17] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12709–12720, 2024.

[18] Paul Green, Wenyang Sun, Wojciech Matusik, and Fredo Durand. Multi-aperture photography. In *Special Interest Group on GRAPHics and Interactive Techniques*, pages 68–es. 2007.

[19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, page 6840–6851, 2020.

[21] Lukas Höllein, Aljaž Božič, Norman Müller, David Novotny, Hung-Yu Tseng, Christian Richardt, Michael Zollhöfer, and Matthias Nießner. ViewDiff: 3D-Consistent Image Generation with Text-To-Image Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

[23] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[24] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[26] Donggoo Jung, Daehyun Kim, and Tae Hyun Kim. Continuous exposure learning for low-light image enhancement using neural ODEs. *International Conference on Learning Representations*, 2025.

[27] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing*, 27(3):1126–1137, 2017.

[28] Jun Ke, Hui Zeng Li, Lei Wang, Zhou Zhang, Weijie Xie, and Hao Zheng. Musiq: Multi-scale image quality assessment neural network. In *IEEE/CVF Conference on International Conference on Computer Vision*, pages 12514–12524, 2021.

[29] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.

[30] Seon Joo Kim and Marc Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE transactions on pattern analysis and machine intelligence*, 30(4):562–576, 2008.

[31] Young-Hwan Kim, Seon Joo Kim, and In So Kweon. A new inverse tone mapping method for hdr images using the image signal processing pipeline. In *European Conference on Computer Vision*, pages 206–220, 2012.

[32] Dehong Kong, Fan Li, Zhixin Wang, Jiaqi Xu, Renjing Pei, Wenbo Li, and WenQi Ren. Dual prompting image restoration with diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12809–12819, 2025.

[33] Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing Text-to-Image Diffusion with Object Viewpoint Control. In *Special Interest Group on GRAPHics and Interactive Techniques Asia*, 2024.

[34] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1931–1941, 2023.

[35] Fan Li, Zixiao Zhang, Yi Huang, Jianzhuang Liu, Renjing Pei, Bin Shao, and Songcen Xu. Magiceraser: Erasing any objects via semantics-aware control. In *European Conference on Computer Vision*, pages 215–231. Springer, 2024.

[36] Zhihao Li, Ming Lu, Xu Zhang, Xin Feng, M. Salman Asif, and Zhan Ma. Efficient visual computing with camera raw snapshots. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4684–4701, 2024.

[37] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[39] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5637–5646, 2022.

[40] Andrew Marmon, Grant Schindler, José Lezama, Dan Kondratyuk, Bryan Seybold, and Irfan Essa. CamViG: Camera aware image-to-video generation with multimodal transformers. *arXiv preprint arXiv:2405.13195*, 2024.

[41] Fan Chen Xiaoying Xing Longyin Wen Chen Chen Sijie Zhu Ming Li, Xin Gu. Superedit: Rectifying and facilitating supervision for instruction-based image editing. *arXiv preprint arXiv:2505.02370*, 2025.

[42] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

[43] Hariharan Nagasubramaniam and Rabih Younes. Bokeh effect rendering with vision transformers. *Authorea Preprints*, 2022.

[44] Ren Ng. Light field photography with a hand-held plenoptic camera. In *Stanford Tech Report CSTR*, 2005.

[45] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When neural rendering meets classical rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[46] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. MPIB: An mpi-based bokeh rendering framework for realistic partial occlusion effects. In *European Conference on Computer Vision*, 2022.

[47] Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. *International Conference on Learning Representations*, 2024.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[49] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

[50] Jingjing Ren, Wenbo Li, Haoyu Chen, Renjing Pei, Bin Shao, Yong Guo, Long Peng, Fenglong Song, and Lei Zhu. Ultrapixel: Advancing ultra high-resolution image synthesis to new peaks. In *Advances in Neural Information Processing Systems*, 2024.

[51] Jingjing Ren, Wenbo Li, Zhongdao Wang, Haoze Sun, Bangzhen Liu, Haoyu Chen, Jiaqi Xu, Aoxue Li, Shifeng Zhang, Bin Shao, et al. Turbo2k: Towards ultra-efficient and high-quality 2k video synthesis. *IEEE/CVF Conference on International Conference on Computer Vision*, 2025.

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10674–10685, 2022.

[53] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[55] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *Advances in Neural Information Processing Systems*, 36:39443–39469, 2023.

[56] Tim Seizinger, Florin-Alexandru Vasluianu, Marcos V Conde, and Radu Timofte. Bokehlicious: Photorealistic bokeh rendering with controllable apertures. *arXiv preprint arXiv:2503.16067*, 2025.

[57] Yichen Sheng, Zixun Yu, Lu Ling, Zhiwen Cao, Xuaner Zhang, Xin Lu, Ke Xian, Haiting Lin, and Bedrich Benes. Dr. Bokeh: Differentiable occlusion-aware bokeh rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4515–4525, June 2024.

[58] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[59] Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, 37(4):1–13, 2018.

[60] Ning-Hsu Wang, Ren Wang, Yu-Lun Liu, Yu-Hao Huang, Yu-Lin Chang, Chia-Ping Chen, and Kevin Jou. Bridging unsupervised and supervised depth from focus via all-in-focus supervision. In *IEEE/CVF Conference on International Conference on Computer Vision*, 2021.

[61] Yujie Wang, Praneeth Chakravarthula, and Baoquan Chen. Dof-gs: Adjustable depth-of-field 3d gaussian splatting for refocusing, defocus rendering and blur removal. *arXiv preprint arXiv:2405.17351*, 2024.

[62] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7677–7689, 2023.

[63] Takashi Watanabe and Shree K. Nayar. Depth from defocus: A real aperture imaging approach. *International Journal of Computer Vision*, 122(2):151–173, 2016.

[64] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2024.

[65] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. CamCo: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.

[66] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[67] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[68] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Zhang Di, Xiaodong Chen, and Jing Liao. Direct-a-Video: Customized video generation with user-directed camera movement and object motion. In *Special Interest Group on GRAPHics and Interactive Techniques*, page 12, 2024.

[69] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. *arXiv preprint arXiv:2411.15738*, 2024.

[70] Yu Yuan, Xijun Wang, Yichen Sheng, Prateek Chennuri, Xingguang Zhang, and Stanley Chan. Generative photography: Scene-consistent camera control for realistic text-to-image synthesis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[72] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 in Section 1 clearly state the main claims.

   Guidelines:

   - The answer NA 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 NA 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: Please refer to Section 6.

Guidelines:

- The answer NA 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: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA 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: We clearly describe the implementation details of our method in Section 5.

   Guidelines:

   - The answer NA 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: [Yes]

   Justification: The code and data will be made publicly available upon acceptance of the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We clearly describe the training and test details of our method in Section 5.

   Guidelines:

   - The answer NA 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: We do not report error bars because our method demonstrates consistently significant improvements, as shown in Table 1 and Table 2. Additionally, the scale of experiments and associated computational cost make repeated trials impractical.

   Guidelines:

   - The answer NA 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: [Yes]

   Justification: The detailed description is presented in Section 5.

   Guidelines:

   - The answer NA 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: We have carefully checked the Ethics Guidelines to make sure our research is with it

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: Please refer to Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

Justification: This paper poses no such risks.

Guidelines:

- The answer NA 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: [NA]

Justification: We do not use existing assets.

Guidelines:

- The answer NA 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: [Yes]

Justification: We introduce a new dataset, CamEdit50K, for camera-aware image editing with continuous parameter annotations. The dataset includes both real and synthetic image pairs with aligned camera parameter instructions.

Guidelines:

- The answer NA 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: [NA]

Justification: This work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: GPT-4o was only used for evaluating editing results. It is not a component of our core methodology and does not affect the scientific rigor or originality of the research.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.