Language-Free Generative Editing from One Visual Example

Omar Elezabi Eduard Zamfir Zongwei Wu* Radu Timofte
Computer Vision Lab, CAIDAS & IFI, University of Würzburg

Abstract

Text-guided diffusion models have advanced image editing by enabling intuitive control through language. However, despite their strong capabilities, we surprisingly find that SOTA methods struggle with simple, everyday transformations such as rain or blur. We attribute this limitation to weak and inconsistent textual supervision during training, which leads to poor alignment between language and vision. Existing solutions often rely on extra finetuning or stronger text conditioning, but suffer from high data and computational requirements. We argue that diffusion-based editing capabilities aren't lost but merely hidden from text. The door to cost-efficient visual editing remains open, and the key lies in a vision-centric paradigm that perceives and reasons about visual change as humans do, beyond words. Inspired by this, we introduce ***Visual Diffusion Conditioning*** *(VDC), a training-free framework that learns conditioning signals directly from visual examples for precise, language-free image editing. Given a paired example—one image with and one without the target effect—VDC derives a visual condition that captures the transformation and steers generation through a novel condition-steering mechanism. An accompanying inversion-correction step mitigates reconstruction errors during DDIM inversion, preserving fine detail and realism. Across diverse tasks, VDC outperforms both training-free and fully fine-tuned text-based editing methods. The code and models are open-sourced at* `omaralezaby.github.io/vdc/`

1. Introduction

Diffusion models have revolutionized visual synthesis, powering the current state-of-the-art in image editing [9, 14, 45]. Notably, text-guided diffusion models enable intuitive manipulation through natural language prompts [8, 38, 57], offering strong spatial and semantic control.

Despite their impressive flexibility, we find that current text-guided diffusion models often struggle with simple visual transformations such as rain, haze or blur. As we illustrate in Fig. 1, their internal representations fail to match the semantics of these textual descriptions. We link this behavior to weak and inconsistent supervision: diffusion models rely on image–caption pairs and thus learn only the concepts explicitly described in the training data. Consequently, visual phenomena that are rarely or ambiguously captioned exhibit poor alignment between text prompts and their associated visual features.

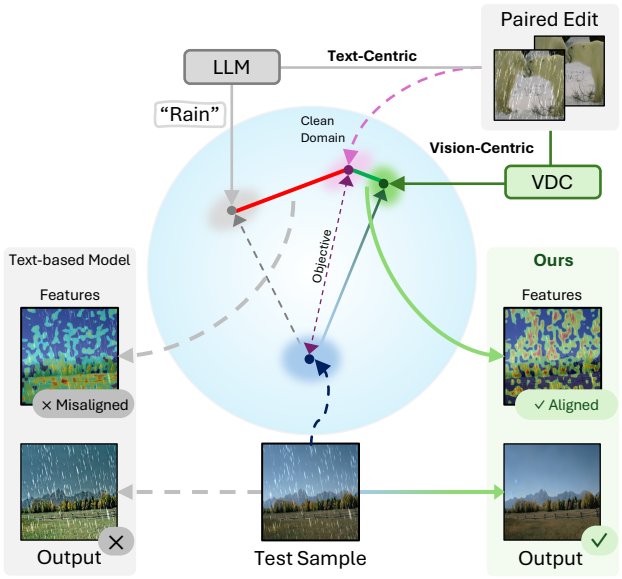


Figure 1. *Text–image misalignment in diffusion latent space.* Text-guided generative models rely on language, which often fails to capture appearance-level transformations, *e.g.* rain, leading to semantic but visually *misaligned* directions. Our method, **Visual Diffusion Conditioning** (VDC), instead learns a vision-centric conditioning signal directly from paired visual examples, uncovering the *correct* transformation direction within the latent space. By steering the diffusion process along this aligned path, VDC achieves faithful and realistic edits, bridging the gap between text semantics and visual representations.

A natural solution might be to fine-tune the model for these missing concepts [3, 26, 56, 63]. However, retraining large diffusion models is computationally intensive and data-hungry, rendering it impractical for most editing scenarios. Importantly, diffusion models already encode rich, structured visual representations that extend beyond their textual supervision [22]. The limitation arises from weak language-vision alignment, which obscures access to the

*Corresponding Author

full visual manifold, as shown in Fig. 2. We propose to bridge this gap through a *vision-centric* perspective on editing. Rather than relying on language to approximate visual intent, our approach treats manipulation as a process grounded in perceptual change. Visual examples—unlike text—can unambiguously express such changes: a pair of images naturally encodes degradations, or stylistic variations that are difficult to capture verbally. By extracting conditioning signals directly from visual examples, we can translate observable differences into latent-space directions that operate on the model's existing visual representations, *c.f.* Fig. 2. This motivates an editing framework driven by *visual exemplars* instead of language.

Building on this, we introduce **Visual Diffusion Conditioning** (VDC), a training-free diffusion editing framework that learns visual conditioning signals from example image pairs. Instead of text prompts, VDC derives a compact representation that encodes the transformation between two visual domains (*e.g.*, clean $\leftrightarrow$ degraded). Once extracted, this visual condition can be transferred to unseen images, enabling consistent and controllable edits. Prior training-free methods [4, 13, 16, 21, 29, 32–34, 48, 50–52, 58] typically operate by inverting the diffusion process and modifying latent trajectories through textual guidance. While effective for semantic manipulation, they remain limited by language–vision misalignment and struggle to express fine-grained, appearance-level changes. Besides, current exemplar-driven approaches [11, 19, 35, 47, 53] partially address this issue by defining edits from image pairs, but most rely on pretrained vision–language models [37] or additional finetuning [53], which reduces generality and increases computational cost.

In contrast, our VDC framework introduces pure visual conditioning, leveraging the pretrained latent structure. Our framework builds on two core components: (i) a *condition steering mechanism* that modulates the sampling process via posterior score guidance [46], enabling precise and stable edits without retraining; and (ii) an *inversion correction step* that compensates for error accumulation in DDIM inversion [9, 45], preserving perceptual quality. In summary, our main contributions are:

- A diffusion editing framework, termed ***Visual Diffusion Conditioning***, that learns directly from visual examples.
- A stable, lightweight neural embedding that captures edit semantics from a single example pair, enabling training-free yet generalizable editing.
- A sampling and inversion strategy that achieves precise editing while preserving perceptual fidelity.

2. Related Works

Text-based Image Editing. Instruction-based image editing methods [3] were proposed to modify an input image according to text instructions. These approaches typically employ generative models to synthesize large-scale instruction-based editing datasets, which are then used to fine-tune diffusion models for conditional image editing. Subsequent works refined this paradigm by curating higher-quality datasets [61] and leveraging improved architectures and generative backbones [23, 26, 42, 56, 63]. To reduce the dependence on large-scale instruction data and computationally expensive training, train-free methods [4, 13, 16, 18, 21, 29, 32–34, 48, 50–52, 58] were introduced. These methods exploit the intrinsic generative and semantic capabilities of pretrained text-to-image (T2I) diffusion models to perform edits without retraining. They typically invert the diffusion process [9, 16, 45, 52, 58] to recover the latent noise representation of an input image, and then modify conditioning components such as the textual prompt [18, 33, 34], self-attention maps [4, 48, 50], or cross-attention modules [13, 21] to realize desired edits. Despite their flexibility, purely text-based methods often struggle to capture fine-grained or compositional edits that go beyond what can be easily expressed in language.

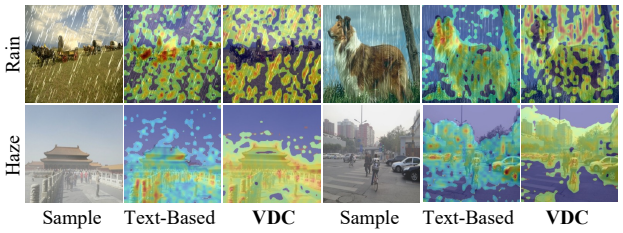
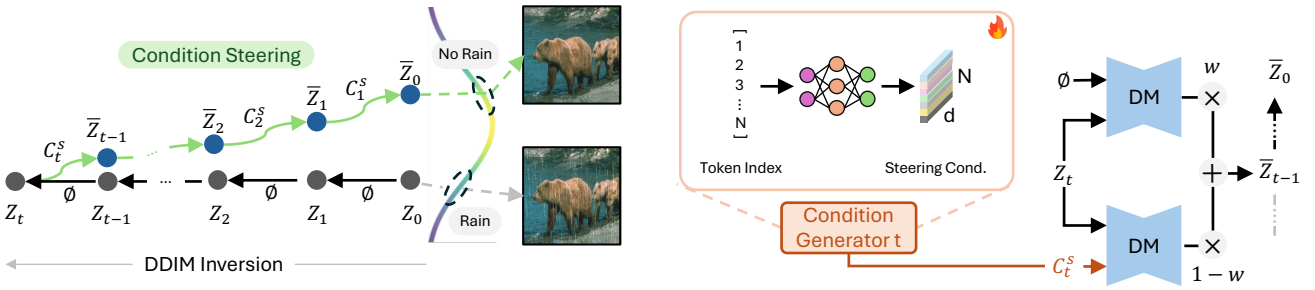


Figure 2. *Language-Vision misalignment.* The internal representations of LDM [38] fail to accurately capture the semantics of degradations such as "rain" or "haze". Attention maps under text-based conditioning remain object-centric and do not correspond to degradation-specific visual attributes. Our VDC framework realigns attention focus toward true visual cues, recovering meaningful features that correspond to rain streaks and hazy regions.

Exemplar-based Editing. A core limitation of text-based editing lies in its reliance on natural language, which is often ambiguous and insufficient for describing complex, localized, or stylistic edits. To address this, visual exemplar-based editing methods incorporate visual examples to define edits more precisely [11, 19, 35, 47, 53]. These approaches learn from pairs of "before" and "after" example images to infer a transformation that can be applied to new inputs. Typically, they employ textual or joint vision–language representations to model the relationship between the visual example pair and the input image. However, even these methods depend on text-aligned latent spaces, inheriting the limitations of T2I diffusion models, such as the imperfect alignment between textual embeddings and visual features. Although some works attempt to fine-tune diffusion models directly for the visual instruction setting, they still rely on VLMs [37] to extract edit semantics [31, 53]. This dependence often leads to the loss of global context or fine visual details, constraining edit fidelity and controllability.



(a) *DDIM inversion with Condition Steering.* (b) *Steering Condition Generation.*

Figure 3. *Proposed VDC framework.* (a) Given a real image, we first invert it through DDIM and apply the learned steering condition C_t^s to guide sampling toward the desired visual feature (e.g., removing rain) while preserving content and quality. (b) A lightweight *Condition Generator* produces per-step steering embeddings from token indices, representing the target visual feature. These conditions modulate the diffusion outputs through weighted score blending, enabling training-free visual editing without textual prompts.

Diffusion for Inverse Problems. Diffusion models have also been successfully applied to inverse problems [5, 6, 10, 54, 64] due to their powerful ability to model complex data distributions. By reformulating image restoration as a guided sampling task, diffusion models can recover clean images that correspond to a given degraded observation—achieving zero-shot restoration without additional training. Initially introduced for image-space diffusion models, these approaches were later extended to latent diffusion models to better exploit their semantic priors and efficiency [20, 39, 40, 44, 55, 60]. Nevertheless, these methods typically assume known degradation operators (*e.g.*, blur kernels, noise levels), which limits their generalization to complex, spatially varying degradations such as haze, rain, or reflection removal.

3. Methodology

Diffusion Preliminaries. Diffusion models generate data by iteratively denoising a latent variable z_t sampled from a Gaussian prior. At each timestep t, a noise prediction network $\epsilon_\theta(z_t, t, C)$ estimates the denoised sample conditioned on C, which can be a text embedding or other guidance signal. Inversion methods such as DDIM [45] allow reconstructing a latent trajectory from a real image, enabling editing in latent space. As shown in Fig. 3a, our framework builds on these foundations by replacing textual conditioning with a learned *visual condition*, used to steer the generative process toward appearance-level transformations.

3.1. Editing by Visual Conditioning

VDC builds on the observation that diffusion models implicitly recognize visual features even when these features lack corresponding textual representations. Although text prompts fail to access such features, they can be revealed by shifting from language-based to purely visual conditioning. We achieve this by identifying a condition that captures a specific transformation through visual examples. Given an image pair before and after editing, (R_B, R_A), we derive a visual condition C^s that encodes the transformation within the model's learned data distribution, as shown in Fig. 3b. By inverting a real image and applying this condition during the generative process, we steer the model to reproduce the desired edit. This enables representation and manipulation of visual features without textual prompts, unlocking the full expressive capacity of the diffusion latent space.

Algorithm 1 Steering Condition Generator Optimization

Input: R_B Visual Example Before Editing, R_A Visual Example After Editing, $Itrs$ Number of optimization iterations, $p \sim [T, 0)$ Diffusion step for resampling start, N Number of tokens in the condition, ϕ Null Condition, E and D Encoder and Decoder respectively.
Output: Optimized Steering Condition C^s

$Z^B, Z^A = E(R_B), E(R_A)$
▷ partial inversion to step p
$Z_p^B = \text{DDIM}_{\text{Inversion}}(Z^B, t = (1, ...p), \phi)$
for $i = 1, ...Itrs$ **do**
  for $t = p, ...1$ **do**
    ▷ generate step condition
    $C_t^s = \text{MLP}_t(1, ...N)$
    $\epsilon_{init} = \epsilon_\theta(Z_t^B, t, \phi)$ ▷ adjust sampling
    $\epsilon_{\text{steering}} = \epsilon_\theta(Z_t^B, t, C_t^s)$
    $\hat{\epsilon} = (1 - w) * \epsilon_{\text{steering}} + w * \epsilon_{\text{init}}$
    $Z_{t-1}^B = \text{DDIM}_{\text{Step}}(Z_t^B, t, \hat{\epsilon})$
  end for
  ▷ Optimize Condition Generator
  $\mathcal{L} = ||Z_0^B - Z^A||_2^2 + ||D(Z_0^B) - R_A||_2^2$
  $\text{MLP}_{1,...t} = \text{MLP}_{1,...t} + \text{AdamGrad}(\mathcal{L})$
end for
return $C^s \leftarrow \text{MLP}_{1,...t}(0, ...N)$

3.2. Condition Steering

To completely detach from the textual space, we consider an unconditional generative process and manipulate the image by steering the sampling trajectory using a condition that represents the visual feature to be edited or removed (*e.g.*, rain, fog, or noise). Given a condition representing a visual

Algorithm 2 DDIM Inversion Correction

Input: Z_0 latent to be inverted , I number of iterations, $p \sim [T, 0)$ Diffusion resampling start, ϕ Null Condition
Output: Corrected Noised Latent z_p^*
$\bar{Z}_p = \text{DDIM}_{\text{Inversion}}(Z_0, t = (1, ...p), \phi)$
  for $i = 1, ...I$ **do**
    $\hat{Z}_0 = \text{DDIM}_{\text{Forward}}(\bar{Z}_p, t = (p, ...1), \phi)$
    $\mathcal{L} = ||\hat{Z}_0 - Z_0||_2^2$
    $\bar{Z}_p = \bar{Z}_p - \text{AdamGrad}(\mathcal{L})$
  end for
return $\bar{Z}_p$

feature C^s, we steer the generative process according to the posterior score function [46] of the unconditional model:

$$\nabla_x \log p(x|C^s) = \nabla_x \log p(x) + \nabla_x \log p(C^s|x) \quad (1)$$

For tasks such as deraining or dehazing, where C^s denotes the feature to be removed, the goal is to steer sampling away from the high-density region of that feature. This can be expressed as the posterior score function for $-C^s$:

$$\nabla_x \log p(x|-C^s) = \nabla_x \log p(x) - s * \nabla_x \log p(C^s|x) \quad (2)$$

Here, s is a hyperparameter controlling the steering intensity, and by Bayes' rule, $p(C^s|x) \sim p(x|C^s)/p(x)$. Expanding this relation gives:

$$\nabla_x \log p(x| - C^s) = \nabla_x \log(x) - s * (\nabla_x \log p(x|C^s) - \nabla_x \log p(x)) \quad (3)$$

Adapting this to the noise prediction model in LDM, where $\nabla_x \log(x) \sim \epsilon_\theta(z_t, t, \phi)$ and $\log p(x|C^s) \sim \epsilon_\theta(z_t, t, C_t)$, we can rewrite the formulation as:

$$\begin{aligned}\epsilon_\theta(z_t, -C^s) &= \epsilon_\theta(z_t, \phi) - s * (\epsilon_\theta(z_t, C^s) - \epsilon_\theta(z_t, \phi)) \\ &= \epsilon_\theta(z_t, C^s) + (1 + s) * (\epsilon_\theta(z_t, \phi) - \epsilon_\theta(z_t, C^s)) \\ &= (1 - w) * \epsilon_\theta(z_t, C^s) + w * \epsilon_\theta(z_t, \phi) \quad (4)\end{aligned}$$

where $w = 1 + s$. This formulation enables direct manipulation of the visual feature represented by C^s by steering the trajectory of the unconditional generative process used to invert the real image. Editing the image in this way avoids generative artifacts, since we update the inverted image ($out = Z(\phi) + Z(C_\theta)$) rather than generating a new image ($out = Z(C_\theta)$), analogous to a global residual connection in image-to-image networks [36]. We visualize this process in Fig. 3a.

3.3. Condition Representation for Visual Features

In diffusion models, the conditioning input is typically represented as a sequence of tokens, each corresponding to an encoded word in the textual prompt (*e.g.*, Stable Diffusion [38] accepts up to 77 tokens as input). Optimizing textual embeddings has been used to improve diffusion inversion of real images [34] or to personalize the generative process by learning an embedding for a specific object [41]. This optimization treats text embeddings as trainable parameters and updates them according to a chosen objective function. However, the process depends on an initial prompt embedding and is often unstable, allowing optimization of only a small number of tokens [34, 47].

To fully remove textual dependency, we generate a new embedding directly from a condition generator network. Inspired by Implicit Neural Representations (INR) [43, 49], which encode images as continuous functions over pixel coordinates, we represent the visual edit condition as a continuous function over token indices. Specifically, we employ a lightweight three-layer MLP and, following INR literature, apply Fourier features to the input indices to improve expressiveness [49]. This formulation provides stable optimization when learning the steering condition that represents a desired edit. The improved stability allows optimization of all 77 tokens, enabling full access to the model's visual condition space. Further, since each token is generated from a continuous function conditioned on token indices, the network naturally establishes communication across tokens, producing smooth and coherent condition representations. For finer control during editing, we optimize a separate condition generator for each diffusion step.

$$\begin{aligned}C_t^s &= \text{MLP}_t(1, ...N) \\ \min_{\text{MLP}_t} &||Z_{t-1}^* - Z_{t-1}(Z_t, t, C_t)||_2^2 \quad (5)\end{aligned}$$

3.4. Optimization and Inversion Refinement

Previous condition optimization methods typically optimize the condition using the output of a single diffusion step [34]. However, this approach forces most edits to occur during the early diffusion steps, leaving the later stages primarily for refinement. In contrast, our VDC optimizes all condition generators jointly based on the final output after the complete diffusion process. This formulation allows the model to decide how edits are distributed across the diffusion trajectory, rather than concentrating them in the initial steps. Accordingly, the optimization in 5 becomes:

$$\begin{aligned}C_{p,...1}^s &= \text{MLP}_{1,...p}(1, ...N) \\ \min_{\text{MLP}_{1,...p}} &||Z_0^* - Z_0(Z_p, t = (p, ...1), C_{p,...1})||_2^2 \quad (6)\end{aligned}$$

Here, N is the number of tokens in the condition, $p \sim [T, 0)$ denotes the starting step of the partial diffusion process, and $t \sim [p, 0)$ represents the current diffusion step. z_0^* is the ground-truth latent, while z_0 is the model output obtained using the optimized steering condition $C_{p,...1}$. This formulation provides the model with greater flexibility to adapt the applied edits dynamically at each diffusion step.

Inversion Correction. DDIM inversion assumes that $Z_{t-1} \sim Z_t$, meaning that adjacent diffusion steps are nearly

Table 1. *Comparison to state-of-the-art image editing.* FID (↓) and LPIPS(↓) are reported on the full RGB images. Our method sets a new state-of-the-art on average across all benchmarks. '-' represents unreported results. The **best** performances are highlighted.

Type	Method	SR		DeBlur		DeNoise		DeRain		DeHaze		Colorization	
		FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
T-Edit	P2P [13]	126.47	0.6662	45.62	0.5220	142.95	0.5593	139.19	0.3122	44.09	0.2183	121.87	0.2931
	Null-Opt [34]	73.48	0.5510	51.89	0.5258	160.88	0.6059	167.61	0.5050	91.76	0.4917	197.81	0.5881
	Negative-Cond [33]	63.22	0.4807	43.61	0.4528	96.19	0.4764	118.76	0.3157	43.20	0.2193	135.63	0.3407
I-Edit	Instruct-Pix2Pix [3]	92.79	0.5828	142.91	0.7081	155.12	0.6298	179.93	0.4285	36.42	0.2399	115.74	0.2975
	OmniGen [56]	59.66	0.4596	46.18	0.4188	150.80	0.4663	119.87	0.3081	42.77	0.2169	134.43	0.3438
	SuperEdit [26]	89.07	0.5481	56.22	0.5307	172.50	0.5866	185.98	0.4489	49.27	0.2960	116.37	0.3860
	ICEdit [63]	50.14	0.4922	45.54	0.4734	128.55	0.5385	149.44	0.3300	170.11	0.5961	104.72	0.2882
Zero-IR	PSLD [39]	**31.90**	0.2839	42.89	0.3683	115.17	0.3660	-	-	-	-	202.71	0.6242
	TReg[20]	49.15	0.5161	52.07	0.4379	94.11	0.5392	-	-	-	-	183.27	0.7713
	DAPS[60]	47.14	0.3290	59.85	0.3413	148.42	0.4137	-	-	-	-	213.36	0.6266
IE-Edit	VISII [35]	110.39	0.4949	122.63	0.5465	248.79	0.8341	203.83	0.5011	198.69	0.6756	298.10	0.5402
	Analogist [11]	83.88	0.4779	75.06	0.4692	143.62	0.5599	158.29	0.6006	68.02	0.3988	156.28	0.5779
	EditClip [53]	77.64	0.5558	78.75	0.5114	99.00	0.5470	174.93	0.3809	44.69	0.2241	138.34	0.3008
VDC	One-Shot	41.41	0.2666	**35.51**	0.2654	89.51	0.2801	87.12	0.2559	35.52	0.1633	107.70	0.2908
	Multi-Shot	45.89	0.2654	42.62	0.2651	88.58	0.2846	69.52	0.2214	34.18	0.1584	107.80	0.2744
	MS+Inverse-Correction	45.00	**0.2624**	41.09	**0.2593**	**82.57**	**0.2768**	**66.92**	**0.2155**	**33.23**	**0.1560**	**105.26**	**0.2729**

identical. However, this assumption holds only for infinitesimally small step sizes, and in practice, it introduces accumulated inversion errors across the diffusion trajectory. To improve inversion accuracy, we propose a refinement method for DDIM inversion. We first perform DDIM inversion up to the desired diffusion step to obtain the initial noised latent z_p. Next, we apply the forward diffusion process using the inverted latent to compute the reconstruction error. Finally, we update the noised latent z_p through gradient-based optimization to minimize this inversion error. The full procedure is summarized in Algorithm 2.

Loss. Since our approach relies on visual examples, we convert the ground-truth image to the latent space and compute the loss directly in that domain. This avoids the disparity between pixel and latent spaces [39], where multiple images may correspond to the same latent representation. However, encoding an image into latent space can result in the loss of fine spatial details, producing inaccurate or overly smoothed edits. To address this, we additionally compute a pixel-space loss by decoding the diffusion latent output back to the image domain. Combining both latent and pixel losses helps preserve spatial fidelity while maintaining semantic consistency during editing:

4. Experiments

We conduct experiments across diverse editing and restoration tasks, comparing against works that adapt T2I diffusion models under different input modalities, training regimes, and optimization strategies. For fairness, we use the same diffusion backbone and include instruction-based models explicitly trained for image editing.

Implementation Details. VDC builds on Stable Diffusion v1.4 [38] with DDIM sampling [45] using 100 steps, operating only on the last 10 steps of the trajectory. The condition generator (CG) is a three-layer MLP network with dimensionality 128. We optimize CG with Adam (β_1=0.9, β_2=0.999) for 200 iterations (batch size 4) using a cosine-annealed learning rate decaying from 5×10^{-3} to 1×10^{-3} [30]. The one-shot setup uses single visual example with flip, rotation, and color-jitter augmentations, while the multi-shot setup increases to eight examples. Condition steering is set to a scale of 7. All experiments run on a single RTX 4090 GPU. We use identical settings across architectures (e.g., SD [38] vs. SANA [57]) and tasks.

Datasets. For super-resolution and deblurring, we use 1K FFHQ [17] samples following DPS [7] degradation. We choose BSD400 [2] testset σ=25 for denoising. For deraining and dehazing, we evaluate on Rain100L [59] and SOTS [24], respectively. For colorization, we convert DIV2K [1] to grayscale. We randomly pick one image per dataset as reference for works requiring visual examples.

Baselines. *Text-edit* (T-Edit) methods manipulate the generation prompt without retraining. We use BLIP [25] to generate captions (*e.g.*, "photo of 3 bears in rain" → "photo of 3 bears") as editing prompts. *Instruction-edit* (I-Edit) methods are trained for text-instruction-based editing; we craft task-specific prompts (*e.g.*, "Remove rain from the image"). *Zero-shot image restoration* (Zero-IR) methods address inverse problems using diffusion priors; we follow DPS [7] for degradation settings. *Image-example* (IE-Edit) methods transfer edits from a reference image to a target; we use the same visual examples as our method for fair comparison. Please refer to the supplementary for more details.

4.1. Comparison to State-of-the-Art Methods

In Tab. 1, VDC surpasses all prior approaches using only a *single* visual example. Its language-free design provides

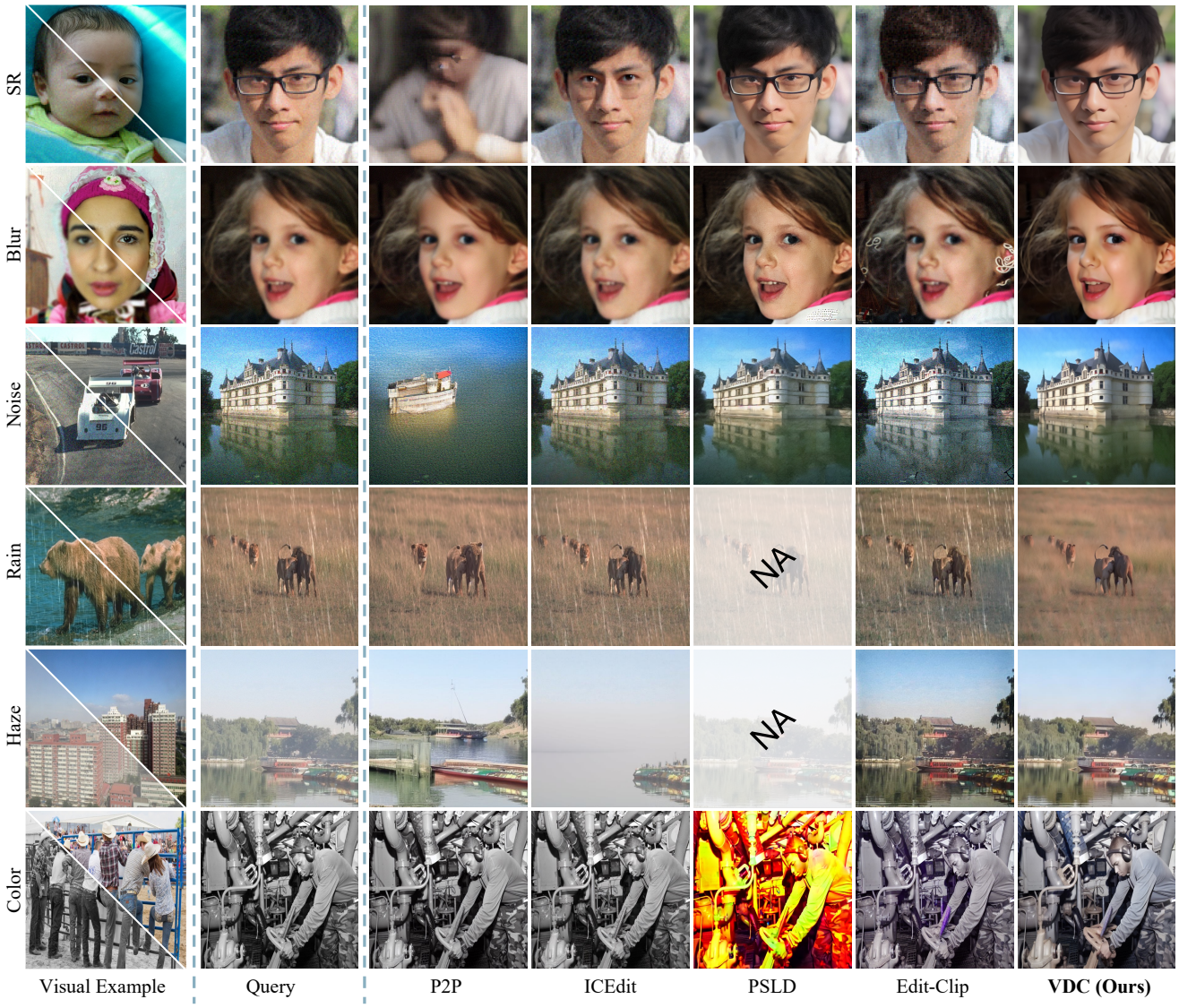


Figure 4. *Visual comparison.* Text- and example-based methods struggle with complex edits due to misalignment or degradation priors. Our one-shot VDC (shown results) yields clean results, with multi-shot and correction modules improving generalization and fidelity.

stronger conditioning than text, overcoming the misalignment that limits text-based methods. IE-Edit methods underperform due to their reliance on joint text–image embeddings, while Zero-IR methods perform better but require known degradation kernels, limiting generalization. Additionally, we compare to diffusion fine-tuning methods like ControlNet [62] and LoRA [15] in Tab. 4, showing their ineffectiveness under low-data regime. Adding more visual examples further improves performance, especially for tasks with diverse degradations (see Sec. 4.2). VDC effectively captures multiple variations (e.g., rain patterns) within a single optimized condition, though too many examples may cause slight overfitting on less variable tasks. Despite inverting only 10% of the diffusion trajectory, the correction inversion module in the multi-shot setup improves detail preservation, particularly for deraining and denoising. VDC is also efficient, requiring about 30 minutes of condition optimization for peak fidelity (200 optimization steps). However, Fig. 6 shows that VDC outperforms OmniGen in just 10 steps ($\sim$ 2 mins). Additionally, Inference incurs zero overhead (just 10 timesteps), as VDC replaces CFG, leaving latency determined by the underlying diffusion model. *Please refer to supplementary for more results.*

Visual Results. Fig. 4 illustrates that text-based methods (P2P, ICEdit) fail to perform complex edits due to text–visual misalignment, often producing corrupted results. Image-example approaches (Edit-CLIP) show similar issues, as they still depend on textual space. Zero-IR methods generate cleaner outputs but introduce noise or color artifacts and rely on known degradation kernels, re-

Table 2. *Contribution analysis.* The upper half evaluates the impact of each module, while the lower half compares different configurations of the condition generator (CG). **Best** results are bolded; the final setup is highlighted.

	Method	SR		DeRain	
		FID ↓	LPIPs ↓	FID ↓	LPIPs ↓
Modules	w/o Data Augmentation	48.53	0.2958	131.82	0.3352
	w/o Pixel Loss	46.93	0.2881	93.79	0.2723
	w/o Condition Steering	55.08	0.2726	122.20	0.26858
	w/o Condition Generator	44.31	0.2718	106.87	0.2568
CG-Setup	Single/Not-Conditioned	41.74	0.3048	89.82	0.2567
	Single/Step-Conditioned	46.49	0.3135	93.67	0.2585
	Per-Step/Text-Conditioned	56.88	0.3331	119.59	0.2828
	Per-Step/Not-Conditioned	**41.41**	**0.2666**	**87.12**	**0.2559**

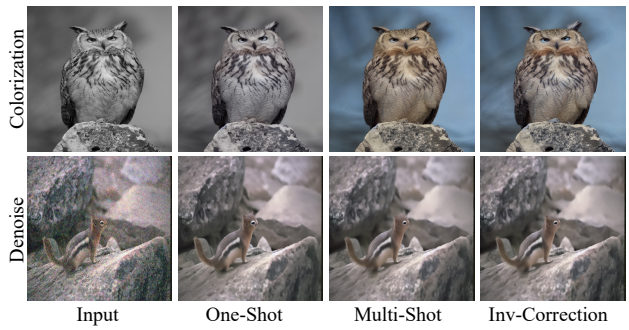


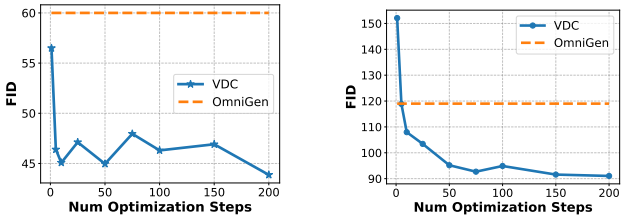
Figure 5. *Number of visual examples.* Increasing the number of examples improves results, especially for tasks with high variability such as colorization. The inversion correction module further enhances detail preservation and overall output quality.

ducing their applicability to tasks such as deraining and dehazing. In contrast, our one-shot VDC accurately captures task-specific visual features, achieving clean, artifact-free results. As shown in Fig. 5, the multi-shot setup generalizes to more complex edits (e.g., colorization), while the correction inversion module further improves fidelity and consistency. Together with the quantitative comparisons, these results showcase the effectiveness of our approach.

4.2. Ablations

We analyze the contribution of each component and design choice in our method, along with insights into diffusion behavior from a conditioning perspective. All experiments use the One-Shot setup on SR and Derain tasks; additional results are in the supplementary material.

Module Contributions. As shown in Tab. 2, we analyze the contribution of each proposed module. *(I) Data augmentation* is crucial in the One-Shot setup, preventing overfitting to the single visual example and improving generalization across diverse patterns. *(II) Pixel loss* substantially enhances quality, as relying solely on latent-space loss discards fine details that the model may misinterpret as edits. *(III) Condition Generator (CG)* implemented as an MLP, improves stability and generalization by generating the full condition jointly rather than optimizing to-



(a) *SR* (b) *DeRain.*

Figure 6. *Optimization trade-off.* VDC outperforms OmniGen in just 10 steps (∼ 2m); extended optimization is optional.

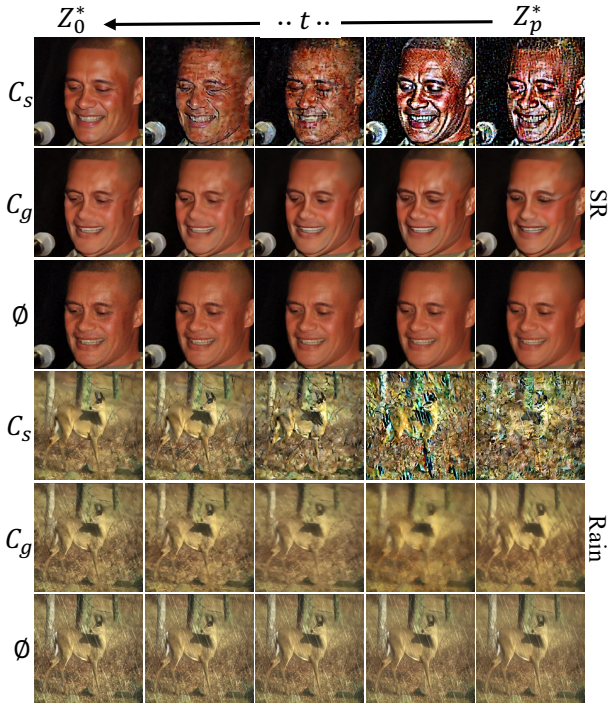


Figure 7. *Condition Steering (C_s) vs. Condition Generation (C_g).* C_s adapts the unconditional path ϕ for the target edit, whereas C_g generates a new image from scratch.

kens independently. *(IV) Condition Steering* provides the largest improvement by optimizing a steering condition that guides the unconditional diffusion trajectory toward the desired edit instead of generating a new image. This focuses the optimization on the edit itself, avoiding entanglement with example content and reducing artifacts. As shown in Fig. 7, our method effectively steers samples within the data distribution toward the target output, producing cleaner and more faithful edits.

Condition Generator Setup. As shown in the lower half of Tab. 2, we evaluate different setups for condition generation. Using a separate condition for each sampling step increases the number of optimization parameters but grants the model greater flexibility, allowing step-specific updates that improve results. We implement this either by feeding the step index as input to a single generator or by assigning a dedicated generator to each step. The latter performs better,

Table 3. *Diffusion path length.* Extending the diffusion path increases variation at the cost of fidelity. **Best** results are bolded; the final setup is highlighted.

Path Length	SR		DeRain	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
5%	56.86	0.3037	89.89	**0.2470**
10%	**41.41**	**0.2666**	**87.12**	0.2559
20%	56.26	0.3134	104.33	0.2708
30%	58.12	0.3193	107.64	0.2856

offering more expressive power and independence across steps. Initializing the generator with a text-based condition, however, reintroduces the text–visual misalignment problem, as the conditioning shifts back into textual space, leading to a notable performance drop. Our final setup employs independent generators for each diffusion step without any textual conditioning. Despite using multiple generators, the additional computational cost is negligible due to the small number of diffusion steps (10) and the compact size of each generator network (approx. 100K parameters).

VDC captures Degradation Attributes. To analyze how visual conditions represent image degradations, we optimized a separate condition for 10 samples per task and visualized them in the condition space in Fig. 8. Conditions from the same task form compact clusters, indicating that similar degradations, such as rain or blur, are consistently encoded as related features, independent of textual representations. This confirms that our visually optimized conditions capture semantic similarities across varying appearances, enabling effective adaptation. Further, in Fig. 13, we report the performance variance across these models. Despite minor fluctuations in complex tasks (DeRain), performance remains robust regardless of the chosen example.

Generalization and Expressiveness. Our method not only learns from synthetic examples but also generalizes to real data and diverse editing scenarios, highlighting the flexibility and scalability of visual conditioning. *(I) Generalization to real data.* Leveraging diffusion priors, VDC generalizes beyond synthetic data, performing well on real images—only eight synthetic samples enable effective deraining on unseen rain patterns (Fig. 9a). *(II) Expressiveness.* Visual examples offer more precise and controllable conditioning. As shown in Fig. 9b, they clearly separate degradations (*e.g.*, haze vs. snow), whereas text prompts often blur this distinction. *(III) Generality.* VDC is model-agnostic and applicable to any conditional diffusion framework, including flow-matching models [28]. Since it learns edits directly from visual examples, it naturally extends beyond pixel-level tasks to broader semantic and object-level modifications. *(IV) Multi-tasking.* Fig. 9b proves a single embedding can learn concurrent tasks (e.g., DeSnow+DeHaze), consolidating multiple needs into one "generalist" solution.

Diffusion Path Length. Following deeper into the diffusion trajectory introduces greater noise to the latent, expanding the output space but increasing deviations from the input image. As visualized in Tab. 3, using longer diffusion paths degrades performance, while overly short paths limit the reachable output space and prevent the desired edit.

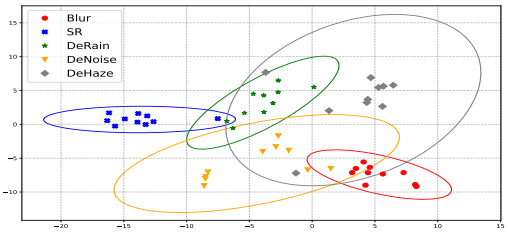


Figure 8. *T-SNE visualization.* Conditions from the same task form clear clusters, showing that similar visual features (*e.g.*, rain, blur) are recognized without textual dependency. This enables one-shot adaptation via condition optimization.

Figure 9. *Generalization & expressiveness.* (a) VDC generalizes from synthetic to real data (RealRain-1K [27]). (b) Visual examples enable fine-grained edits (CDD-11 [12]).

5. Conclusion

Text-guided diffusion models remain limited by weak language–vision alignment, hiding much of their visual editing potential behind text conditioning. *Visual Diffusion Conditioning* (VDC) unlocks this potential by replacing text with visual examples as the source of guidance. VDC learns visual conditions directly from paired examples and steers the diffusion process toward precise, language-free edits through a lightweight condition generator and a condition-steering mechanism. An inversion correction step further preserves fine details and realism.

With as little as one example, VDC adapts text-to-image diffusion models for complex edits such as deraining, deblurring, and dehazing—without retraining or fine-tuning. It achieves accurate, artifact-free results while remaining efficient, training-free, and generalizable to real-world data. Future work could explore extending VDC to unposed or in-the-wild images and studying its behavior on more diverse real-world conditions.

Acknowledgments: This work was supported by The Alexander von Humboldt Foundation.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 5

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 5

[3] 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. 1, 2, 5

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023. 2

[5] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 3

[6] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 3

[7] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 5

[8] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 1

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[10] Ben Fei, Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo, Bo Zhang, and Bo Dai. Generative diffusion prior for unified image restoration and enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9935–9946, 2023. 3

[11] Zheng Gu, Shiyuan Yang, Jing Liao, Jing Huo, and Yang Gao. Analogist: Out-of-the-box visual in-context learning with image diffusion model. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 2, 5

[12] Yu Guo, Yuan Gao, Yuxu Lu, Ryan Wen Liu, and Shengfeng He. Onerestore: A universal restoration framework for composite degradation. In *European Conference on Computer Vision*, 2024. 8

[13] 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. 2, 5

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022. 6

[16] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023. 2

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5

[18] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023. 2

[19] Hyunsoo Kim, Donghyun Kim, and Suhyun Kim. Difference inversion: Interpolate and isolate the difference with token consistency for image analogy generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18250–18259, 2025. 2

[20] Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. *arXiv preprint arXiv:2311.15658*, 2023. 3, 5

[21] Jimyeong Kim, Jungwon Park, Yeji Song, Nojun Kwak, and Wonjong Rhee. Reflex: Text-guided editing of real images in rectified flow via mid-step feature extraction and attention adaptation. *arXiv preprint arXiv:2507.01496*, 2025. 2

[22] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[23] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2

[24] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE transactions on image processing*, 28(1):492–505, 2018. 5

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 5

[26] Ming Li, Xin Gu, Fan Chen, Xiaoying Xing, Longyin Wen, Chen Chen, and Sijie Zhu. Superedit: Rectifying and facilitating supervision for instruction-based image editing. *arXiv preprint arXiv:2505.02370*, 2025. 1, 2, 5

[27] Wei Li, Qiming Zhang, Jing Zhang, Zhen Huang, Xinmei Tian, and Dacheng Tao. Toward real-world single image

deraining: A new benchmark and beyond. *arXiv preprint arXiv:2206.05514*, 2022. 8

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 8

[29] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 289–299, 2023. 2

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[31] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for multi-task image restoration. *arXiv preprint arXiv:2310.01018*, 2023. 2

[32] 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. 2

[33] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2063–2072. IEEE, 2025. 2, 5

[34] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 2, 4, 5

[35] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. *Advances in Neural Information Processing Systems*, 36:9598–9613, 2023. 2, 5

[36] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *IEEE Transactions on Multimedia*, 24:3859–3881, 2021. 4

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[38] 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. 1, 2, 4, 5

[39] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:49960–49990, 2023. 3, 5

[40] Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9472–9481, 2024. 3

[41] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 4

[42] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024. 2

[43] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 4

[44] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. *arXiv preprint arXiv:2307.08123*, 2023. 3

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2, 3, 5

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 4

[47] Adéla Šubrtová, Michal Lukáč, Jan Čech, David Futschik, Eli Shechtman, and Daniel Sỳkora. Diffusion image analogies. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2, 4

[48] Wenhao Sun, Xue-Mei Dong, Benlei Cui, and Jingqun Tang. Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20734–20742, 2025. 2

[49] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020. 4

[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. 2

[51] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023.

[52] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Tam-

ing rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024. 2

[53] Qian Wang, Aleksandar Cvejic, Abdelrahman Eldesokey, and Peter Wonka. Editclip: Representation learning for image editing. *arXiv preprint arXiv:2503.20318*, 2025. 2, 5

[54] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022. 3

[55] Jie Xiao, Ruili Feng, Han Zhang, Zhiheng Liu, Zhantao Yang, Yurui Zhu, Xueyang Fu, Kai Zhu, Yu Liu, and Zheng-Jun Zha. Dreamclean: Restoring clean image using deep diffusion prior. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[56] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025. 1, 2, 5

[57] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 1, 5

[58] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. *arXiv preprint arXiv:2312.04965*, 2023. 2

[59] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020. 5

[60] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905, 2025. 3, 5

[61] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 2

[62] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 6

[63] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025. 1, 2, 5

[64] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1219–1229, 2023. 3