When Test-Time Guidance Is Enough: Fast Image and Video Editing with Diffusion Guidance

Ahmed Ghorbel[1,*] **Badr Moufad**[1,*] **Navid Bagheri Shouraki**[1,5,7] **Alain Oliviero Durmus**[1]
Thomas Hirtz[6] **Eric Moulines**[3,4] **Jimmy Olsson**[8] **Yazid Janati**[2,3,*]

[1]CMAP, Ecole Polytechnique [2]Institute of Foundation Models [3]MBZUAI [4]EPITA
[5]Sorbonne University [6]Lagrange Mathematics and Computing Research Center
[7]EPITA Research Lab [8]KTH Royal Institute of Technology

Abstract

Text-driven image and video editing can be naturally cast as inpainting problems, where masked regions are reconstructed to remain consistent with both the observed content and the editing prompt. Recent advances in test-time guidance for diffusion and flow models provide a principled framework for this task; however, existing methods rely on costly vector–Jacobian product (VJP) computations to approximate the intractable guidance term, limiting their practical applicability. Building upon the recent work of Moufad et al. (2025), we provide theoretical insights into their VJP-free approximation and substantially extend their empirical evaluation to large-scale image and video editing benchmarks. Our results demonstrate that test-time guidance alone can achieve performance comparable to, and in some cases surpass, training-based methods.

1 Introduction

Image and video editing plays a central role in a wide range of applications, including content creation and interactive design. In the era of text-driven generative models, editing tasks can be naturally formalized as *inpainting problems*, where regions of interest to be modified are masked and subsequently refilled with the desired content by manipulating a text prompt. Inpainting is a classical problem in computer vision that has motivated extensive research, with particularly prominent advances arising from recent large-scale generative models based on diffusion and flow matching models (Esser et al., 2024; Batifol et al., 2025; Wu et al., 2025). A direct approach to address inpainting problems is to *train* or *fine-tune* conditional diffusion models to explicitly incorporate additional inputs, such as masks and observed regions, to approximate the corresponding conditional distribution. While effective, these approaches incur non-negligible computational and data costs, which may be prohibitive in many practical scenarios.

An appealing alternative is *test-time guidance*, which formulates inpainting as a Bayesian inverse problem: a pre-trained text-conditional diffusion model defines the prior, while a likelihood enforces consistency with observed regions. Sampling from the resulting posterior yields the desired completion without updating model weights. Despite its flexibility and strong empirical performance (Daras et al., 2024), a central challenge lies in approximating the intractable guidance term. Existing methods use the likelihood evaluated at the model's output, which during sampling, entails repeated vector–Jacobian product computations through the model, which are expensive and limit scalability.

In this work, we build upon the recently introduced VJP-free approximation proposed in Moufad et al. (2025). The core idea is to approximate the oracle guidance term using a mixture in which the intermediate variable of interest is decoupled from the model, thereby eliminating the need for VJP computations. This approximation yields cheap closed-form posterior updates for linear inverse problems, including inpainting tasks. Our contributions are threefold. (i) We shed light on this new

[*] Correspondence: {ahmed.ghorbel, badr.moufad}@polytechnique.edu, yazid.janati@mbzuai.ac.ae

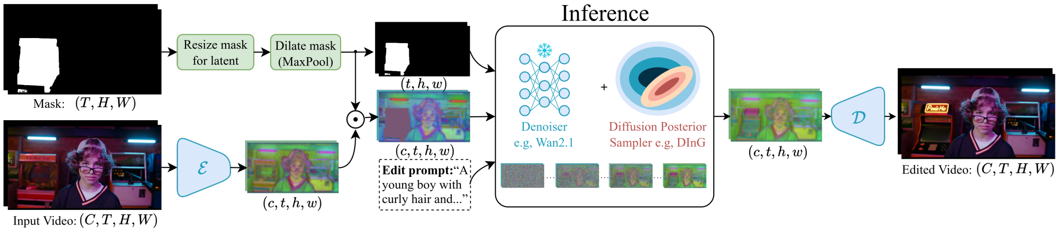


Figure 1: Overview of the editing pipeline for video modalities. The input video and mask are lifted to the latent space for inpainting. A pre-trained and frozen diffusion model is used with a posterior sampler to guide the generation toward prompt-aligned reconstructions, which is then decoded back to pixel space.

approximation and provide theoretical insights on its derivation. (ii) We extend the benchmarks in Moufad et al. (2025) to more image settings as well as video editing tasks, and demonstrate that *test-time guidance alone* can achieve performance comparable to, and in some cases surpass, training-based approaches. (iii) We release an open-source Python package[1] tailored to editing via inpainting, designed to easily accommodate new pre-trained models and training-free samplers. Finally, our results establish test-time guidance as a compelling alternative for image and video editing that requires only black-box access to a pre-trained diffusion model with frozen weights.

2 Editing via Inpaiting with Diffusion Priors

Inpainting as a Bayesian inverse problem. Editing tasks can be framed as inpainting problems, in which regions to be modified are masked and subsequently refilled with new content. Let $x_\star$ denote the reference sample to be edited, and let $\mathbf{M}$ be a binary mask indicating the *observed* regions to be preserved $y = \mathbf{M} \odot x_\star$. This induces the Gaussian likelihood

$$\ell(y \mid x_0) \propto \exp\{-\tfrac{1}{2\gamma^2}\|y - \mathbf{M} \odot x_0\|^2\}, \tag{2.1}$$

where the parameter $\gamma > 0$ promotes stricter consistency with the observed regions, the smaller it is. Since the likehood is agnostic to the values of masked regions, the inpainting problem is inherently ill-posed and the Bayesian formulation works around this by imposing a prior distribution p_0 over plausible reconstructions x_0. Filling the masked regions then reduces to sampling the posterior

$$\pi_0(x_0 \mid y) \propto \ell(y \mid x_0)\, p_0(x_0).$$

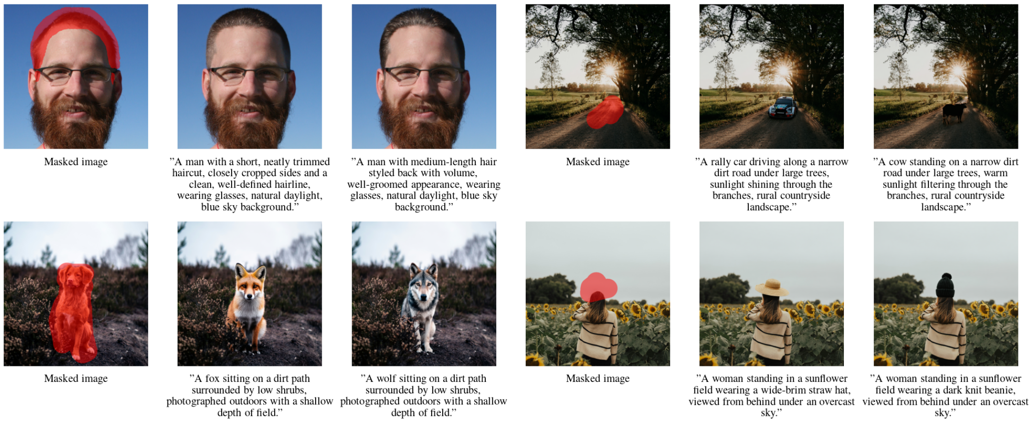
When the prior p_0 is a text-conditional generative model, manipulating the prompt provides a principled way to control the semantics of the refilled regions.

Diffusion Models. A central paradigm in modern generative modeling is learning a sequence of transformations that maps a simple reference distribution p_1, typically a standard Gaussian $\mathcal{N}(0, \mathrm{I})$, to a complex data distribution p_0. Among the many possible constructions of such transformations (Rezende & Mohamed, 2015; Chen et al., 2018; Lipman et al., 2023; Albergo et al., 2023), diffusion models adopt the linear interpolation $X_t = \alpha_t X_0 + \sigma_t X_1$ as a principled inductive bias, where $X_0 \sim p_0$, $X_1 \sim p_1$, and (α_t, σ_t) are deterministic schedules satisfying appropriate boundary conditions. This interpolation induces a family of marginal distributions $(p_t)_{t\in[0,1]}$ that gradually transforms the data distribution p_0 into the Gaussian p_1. Given a discretization $t_0, \ldots, t_K$ of the interval $[0, 1]$, sampling from p_0 is performed by simulating a time-reversed Markov chain: for two consecutive timesteps $(s, t) = (t_k, t_{k+1})$, the following reverse transition is sequentially sampled

$$p^\eta_{s|t}(x_s \mid x_t) = \mathbb{E}\left[q^\eta_{s|0,1}(x_s \mid X_0, X_1) \,\middle|\, X_t = x_t\right], \tag{2.2}$$

where the kernel $q^\eta_{s|0,1}(\cdot \mid X_0, X_1)$ preserves the path marginals $p_s(x_s) = \mathbb{E}[q^\eta_{s|0,1}(x_s \mid X_0, X_1)]$, whereas η controls the stochasticity of the transition (Song et al., 2021a). The transition (2.2) is typically intractable as it depends on the unknown conditional distributions $p_{0|t}$ and $p_{1|t}$. A standard approximation consists of replacing X_0 and X_1 by their conditional expectations given X_t and learning a denoiser $(t, x_t) \mapsto \hat{\mathbf{x}}_0(x_t, t)$ to approximate $\mathbb{E}[X_0 \mid X_t = x_t]$ whereas the expression of the noise predictor $\hat{\mathbf{x}}_1(x_t, t)$ that approximates $\mathbb{E}[X_1 \mid X_t = x_t]$ follows from Tweedie's formula (Roberts & Tweedie), $\hat{\mathbf{x}}_1(x_t, t) = \frac{1}{\sigma_t}\big(x_t - \alpha_t \hat{\mathbf{x}}_0(x_t, t)\big)$. The neural network $\hat{\mathbf{x}}_0$ is then trained by minimizing a simple regression objective (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022).

[1] Link to the code of `DInG-editor` https://github.com/Badr-MOUFAD/ding-editor



Masked image

"A man with a short, neatly trimmed haircut, closely cropped sides and a clean, well-defined hairline, wearing glasses, natural daylight, blue sky background."

"A man with medium-length hair styled back with volume, well-groomed appearance, wearing glasses, natural daylight, blue sky background."

Masked image

"A rally car driving along a narrow dirt road under large trees, sunlight shining through the branches, rural countryside landscape."

"A cow standing on a narrow dirt road under large trees, warm sunlight filtering through the branches, rural countryside landscape."

Masked image

"A fox sitting on a dirt path surrounded by low shrubs, photographed outdoors with a shallow depth of field."

"A wolf sitting on a dirt path surrounded by low shrubs, photographed outdoors with a shallow depth of field."

Masked image

"A woman standing in a sunflower field wearing a wide-brim straw hat, viewed from behind under an overcast sky."

"A woman standing in a sunflower field wearing a dark knit beanie, viewed from behind under an overcast sky."

Figure 2: Editing via inpainting using DING with SD3 as prior. Given masked inputs, the model fills the missing regions according to diverse textual prompts. The runtime is limited to 10 seconds per image (1024px).

Test-time guidance with diffusion priors. As sampling in diffusion models proceeds sequentially via a reverse-time Markov chain, it allows to intervene in-between to bias the transitions in (2.2) so as to target the posterior distribution $\pi_0(\cdot \mid y)$. Seminal works of Song & Ermon (2019); Kadkhodaie & Simoncelli (2020); Kawar et al. (2021) demonstrated that this can be achieved in inference time without additional training. Janati et al. (2025) derived the expression of the *oracle reverse transitions*

$$\pi^{\eta}_{s|t}(x_s \mid x_t, y) \propto \ell_s(y \mid x_s)\, p^{\eta}_{s|t}(x_s \mid x_t), \quad \text{where,}\ \ \ell_s(y \mid x_s) = \mathbb{E}[\,\ell(y \mid X_0) \mid X_s = x_s]\,. \tag{2.3}$$

The intermediate likelihood $\ell_s(y \mid x_s)$ is typically intractable as it involves the conditional $p_{0|s}(\cdot \mid x_s)$. Applying Tweedie's formula yields the *oracle conditional expectation given the observation*,

$$\mathbb{E}[X_0 \mid X_t = x_t, Y = y] = \mathbb{E}[X_0 \mid X_t = x_t] + \alpha_t^{-1}\sigma_t^2\, \nabla_{x_t} \log \ell_t(y \mid x_t), \tag{2.4}$$

which decomposes into two terms: a prior term approximated by the pre-trained prior model $\hat{\mathbf{x}}_0(x_t, t)$, and a second term referred to as the *guidance term*. A widely used approximation introduced in Chung et al. (2023) replaces the intermediate likelihood by a point estimate obtained from the denoiser output, resulting in the tractable approximation $\ell_t(y \mid x_t) \approx \ell\big(y \mid \hat{\mathbf{x}}_0(x_t, t)\big)$.

The VJP-free approximation in Moufad et al. (2025). When combined with Equation (2.4), the approximation of Chung et al. (2023) requires taking a gradient w.r.t. the input of the model, which entails computing a VJP. This becomes a major computational bottleneck, namely, for modern large-scale diffusion models, where VJPs are expensive. To overcome this limitation, Moufad et al. (2025), building on earlier works on VJP-free approximations (Wang et al., 2023; Zhu et al., 2023; Mardani et al., 2024), propose a principled approximation that relies on rewriting the denoiser using the noise predictor as $\hat{\mathbf{x}}_0(x_s, s) = \frac{1}{\alpha_s}\big(x_s - \sigma_s \hat{\mathbf{x}}_1(x_s, s)\big)$, and then replacing the input of $\hat{\mathbf{x}}_1$ by an auxiliary random variable Z_s that has the same marginal as X_s. This yields a mixture approximation

$$\hat{\ell}_s(y \mid x_s) = \mathbb{E}\Big[\ell\Big(y \mid \tfrac{1}{\alpha_s}\big(x_s - \sigma_s \hat{\mathbf{x}}_1(Z_s, s)\big)\Big)\Big]\,, \qquad Z_s \sim p^{\eta}_{s|t}(\cdot \mid x_t), \tag{2.5}$$

which decouples x_s from the model. Based on it, Moufad et al. (2025) directly simulate the posterior transition (2.3) by first sampling z_s and then drawing x_s from the distribution $\hat{\ell}_s(y \mid \cdot, z_s)\, p^{\eta}_{s|t}(\cdot \mid x_t)$. For linear inverse problems, such as inpainting (2.1), the likelihood is Gaussian and linear in x_s, so the second step can be sampled exactly via Gaussian conjugacy (Bishop, 2006).

Theoretical insight. By adding and subtracting Z_s in eq. (2.5), the approximation rewrites as

$$\hat{\ell}_s(y \mid x_s) = \mathbb{E}\Big[\ell\Big(y \mid \hat{\mathbf{x}}_0(Z_s, s) + \tfrac{1}{\alpha_s}(x_s - Z_s)\Big)\Big]\,.$$

This expression can be interpreted as a first-order Taylor expansion of the denoiser $\hat{\mathbf{x}}_0(\cdot, s)$ around Z_s, where the true Jacobian $\nabla \hat{\mathbf{x}}_0(\cdot, s)$ is approximated by the scaled identity $(1/\alpha_s)\mathrm{I}$. Differentiating the relation between the denoiser and the noise predictor yields $\nabla \hat{\mathbf{x}}_0(Z_s, s) = \frac{1}{\alpha_s}\big(\mathrm{I} - \sigma_s \nabla \hat{\mathbf{x}}_1(Z_s, s)\big)$. Hence, the approximation in Moufad et al. (2025) is equivalent to neglecting the Jacobian of the noise predictor. Likewise, the second-order Tweedie formula (Boys et al., 2023), i.e. $\nabla \mathbf{x}^{\theta}_0(Z_s, s) = (\alpha_s/\sigma_s^2)\, \mathbb{C}\mathrm{ov}[X_0 \mid Z_s]$, shows that the same approximation corresponds to assuming the that conditional covariance is isotropic $\mathbb{C}\mathrm{ov}[X_0 \mid Z_s] = (\sigma_s^2/\alpha_s^2)\mathrm{I}$.

Table 2: Quantitative comparison between training-free baselines on `HumanEdit` 1024px and `InpaintCOCO` 512px on image editing tasks using FLUX and SD3 as priors. Runtime is limited to 50 NFEs.

	`HumanEdit`+ FLUX				`HumanEdit`+ SD3				`InpaintCOCO`+ FLUX				`InpaintCOCO`+ SD3			
	FID	pFID	edFID	cPSNR	FID	pFID	edFID	cPSNR	FID	pFID	edFID	cPSNR	FID	pFID	edFID	cPSNR
BLENDED-DIFF	30.8	4.1	25.1	34.6	33.1	8.9	30.6	31.5	41.9	10.4	23.3	28.0	46.0	14.1	27.6	27.2
DDNM	31.2	4.8	31.4	36.4	30.5	4.6	27.7	34.6	45.6	14.2	32.9	30.4	41.6	11.6	26.2	29.8
DIFFPIR	31.1	4.3	26.0	35.4	31.1	4.6	26.4	33.8	43.6	11.9	26.9	28.9	42.2	10.0	22.3	28.5
DING	31.2	5.0	27.5	35.7	30.8	4.7	25.5	34.4	43.2	11.9	26.9	29.9	41.9	10.3	21.9	29.2
FLAIR	30.9	5.2	27.1	35.0	30.7	4.9	31.4	34.2	41.5	15.7	33.1	29.2	42.7	11.8	29.9	29.3
FLOWCHEF	32.3	5.7	29.4	33.5	34.6	9.0	27.9	31.3	52.2	17.0	40.9	29.5	46.6	16.2	28.2	28.0

3 EXPERIMENTS

From pixel- to latent-space inpainting. Modern large-scale diffusion models operate in a compressed latent space defined by an encoder–decoder pair $(\mathcal{E}, \mathcal{D})$. While this design enables scalability to high-resolution data, it complicates training-free guidance, since inverse problems are defined in pixel space whereas guidance is applied in latent space, inducing a nonlinear likelihood through the decoder. For inpainting, however, Avrahami et al. (2023) show that guidance can be performed entirely in latent space by downsampling the binary pixel mask and encoding the input as $x_* = \mathcal{E}(\text{input})$. As illustrated in Figure 1, pixel- and latent-space masking are equivalent despite the encoder nonlinearity. This formulation enables inpainting directly in latent space; particularly, the closed-form update in Equation (2.5) is applicable. We further discuss the limitations of this approach in Appendix B.

Table 1: Quantitative comparison between training-based methods and DING on image editing tasks. Runtime is specified above the baselines.

	FID	pFID	edFID	cPSNR
30 second / image	`HumanEdit` 1024px			
FLUX+ ControlNet	26.8	4.0	24.4	36.5
FLUX Fill	27.9	4.0	22.5	34.1
FLUX+ DING	31.1	4.9	28.1	35.9
SD3 + ControlNet	37.1	11.3	30.6	26.3
SD3 + DING	30.3	4.4	25.4	35.2
10 second / image	`InpaintCOCO` 512px			
FLUX+ ControlNet	43.1	13.6	26.8	31.1
FLUX Fill	41.0	13.4	23.9	28.9
FLUX+ DING	44.1	12.2	28.1	30.1
SD3 + ControlNet	45.8	17.0	27.0	24.4
SD3 + DING	40.9	9.5	20.7	30.3

Experimental Setting. Here, we describe the models, baselines, and datasets used in our experiments and defer the details to Appendix A.

Large-scale text-conditioned priors. For image editing, we consider, in addition to Stable Diffusion 3.5 (SD3) which is exclusively used in Moufad et al. (2025), the large-scale text-to-image model FLUX (Batifol et al., 2025). We also extend the evaluation to video editing tasks: we adopt the text-to-video diffusion models LTX (HaCohen et al., 2024) and Wan2.1 (Wan et al., 2025).

Baselines. We conduct benchmark using both training-free and training-based approaches. Among training-free baselines, we include BLENDED-DIFF (Avrahami et al., 2023), DIFFPIR (Zhu et al., 2023), DDNM (Wang et al., 2023), DING (Moufad et al., 2025), FLAIR (Erbach et al., 2025), and FLOWCHEF (Patel et al., 2024). For training-based methods, we consider controlNet for inpainting methods (Zhang et al., 2023) with SD3 and FLUX as backbones. We also include FLUX Fill, a checkpoint specifically trained for image inpainting. For video inpainting, we compare with Wan2.1-VACE, a checkpoint trained for video inpainting.

Datasets.
We perform image editing experiments on `InpaintCOCO` (Rösch et al., 2024) and `HumanEdit` (Bai et al., 2024). The `InpaintCOCO` dataset contains 1,260 images, while we select a subset of 1,000 images from `HumanEdit`. Both datasets provide editing masks paired with text prompts. For video editing, we use `VPBench` (Bai et al., 2024), which consists of 133 videos, each annotated with editing masks and corresponding text prompts.

Table 3: Quantitative comparison on video editing tasks on `VPBench` dataset with resolution (H, W, T)=$(512, 928, 97)$ All methods are training-free except last row. Runtime is provided above the baselines.

	CLIP	FVD	cPSNR
2min 25s / video	LTX		
BLENDED-DIFF	26.11	0.15	21.82
DDNM	26.09	0.15	23.87
DIFFPIR	26.13	0.15	22.57
DING	25.75	0.16	22.90
FLAIR	26.03	0.15	25.31
FLOWCHEF	25.83	0.19	19.52
5min 25s / video	Wan2.1		
BLENDED-DIFF	26.33	0.14	26.95
DDNM	26.31	0.13	30.00
DIFFPIR	26.36	0.13	28.52
DING	26.24	0.13	29.21
FLAIR	26.30	0.13	29.94
FLOWCHEF	26.17	0.16	25.22
Wan2.1VACE	26.29	0.11	32.69

Evaluation. We consider a compute-constrained setting that mirrors deployment scenarios. When comparing training-free baselines among themselves, we fix the NFEs to 50. For comparisons between training-free and training-based methods, we instead match runtime to ensure a fair assessment; the runtimes are reported in the captions of each table. Our evaluation primarily aims to further validate the approximation in Moufad et al. (2025), which we hence take as reference when benchmarking against training-based methods. For image editing, we report FID and patch FID (pFID) (Chai et al., 2022), the latter providing finer-grained evaluation for high-resolution images. To measure preservation of the unedited content, we compute the context PSNR (cPSNR), defined as the PSNR over the unmasked regions only. To assess the quality of the edited regions, we report the edited-region FID (edFID), obtained by extracting the edited regions and computing pFID over them. For video editing, we report CLIP-Score (Radford et al., 2021), FVD (Unterthiner et al., 2018), and cPSNR. For FID related metrics, lower is better, for CLIP and cPSNR higher is better.

Comments. Quantitative results are summarized in Tables 1 to 3, with qualitative comparisons shown in Figures 4 to 6. Training-free baselines achieve competitive performance across metrics despite not being trained for such tasks. In particular, Table 1 shows that, under the same compute budgets, the training-free method DInG performs on par with, and in even surpasse the training-based method SD3 + ControlNet. Qualitative examples in Figures 4 to 6 further corroborate these findings. It shows high-quality edits and strong controllability for training-free methods, with visual performance comparable to training-based counterparts. Qualitative results on video editing tasks can be found on the project webpage[2].

Python package for editing. We release `DInG-editor`, a Python package for training-free editing via inpainting that implements the pipeline in Figure 1. It currently supports image and video editing with three image priors and two video priors; audio support is under active development. `DInG-editor` is model- and sampler-agnostic: integration of new diffusion priors and training-free samplers is straightforward. The package also provides evaluation scripts, pre-implemented image and video metrics, and comprehensive installation and usage documentation to facilitate reproducibility.

Acknowledgements

The work of Ahmed Ghorbel and Badr Moufad has been supported by Technology Innovation Institute (TII), project Fed2Learn. The work of Eric Moulines has been partly funded by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work was granted access to the HPC resources of IDRIS under the allocations 2025-AD011016497 by GENCI.

[2] https://badr-moufad.github.io/ding-editor/

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42 (4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592450. URL https://doi.org/10.1145/3592450.

Jinbin Bai, Wei Chow, Ling Yang, Xiangtai Li, Juncheng Li, Hanwang Zhang, and Shuicheng Yan. Humanedit: A high-quality human-rewarded dataset for instruction-based image editing. *arXiv preprint arXiv:2412.04280*, 2024.

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

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023.

Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *European conference on computer vision*, pp. 170–188. Springer, 2022.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

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. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

Julius Erbach, Dominik Narnhofer, Andreas Dombos, Bernt Schiele, Jan Eric Lenssen, and Konrad Schindler. Solving inverse problems with flair. *arXiv preprint arXiv:2506.02680*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Yazid Janati, Badr Moufad, Mehdi Abou El Qassim, Alain Durmus, Eric Moulines, and Jimmy Olsson. A mixture-based framework for guiding diffusion models. *preprint*, 2025.

Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1YO4EE3SPB.

Badr Moufad, Navid Bagheri Shouraki, Alain Oliviero Durmus, Thomas Hirtz, Eric Moulines, Jimmy Olsson, and Yazid Janati. Efficient zero-shot inpainting with decoupled diffusion guidance. *arXiv preprint arXiv:2512.18365*, 2025.

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv preprint arXiv:2412.00100*, 2024.

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*, pp. 8748–8763. PmLR, 2021.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

G. O. Roberts and R. L. Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. 83(1):95–110. ISSN 0006-3444. doi: 10.1093/biomet/83.1.95. URL https://doi.org/10.1093/biomet/83.1.95.

Philipp J. Rösch, Norbert Oswald, Michaela Geierhos, and Jindřich Libovický. Enhancing conceptual understanding in multimodal contrastive learning through hard negative samples, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mRieQgMtNTQ.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

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*, pp. 3836–3847, 2023.

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*, pp. 1219–1229, 2023.

Reproducibility statement. All experiments are reproducible using the released Python package `DInG-editor`[3], which includes implementations of all training-free and training-based methods considered in this work, along with the corresponding evaluation scripts.

Ethics statement. Test-time guidance lowers computational barriers for creative media applications, broadening accessibility. However, diffusion-based inpainting methods are inherently dual-use and may be misused to generate deceptive and harmful content, such as manipulated images or synthetic media that obscure authenticity. This risk underscores the importance of responsible deployment, including clear usage guidelines and traceable outputs.

A Details on models and baselines

Large-scale models. For image editing, we use SD3 (Esser et al., 2024) and FLUX (Batifol et al., 2025), both equipped with a VAE featuring a spatial downsampling factor of $\times 8$. For video editing, we consider LTX (HaCohen et al., 2024), which employs a video VAE with spatial and temporal downsampling factors of $\times 32$ and $\times 8$, respectively, and Wan2.1, which uses $\times 8$ spatial and $\times 3$ temporal downsampling. For all models, we use the default guidance scale provided by the authors.

Training-based methods. We include baselines that are explicitly trained for inpainting. For images, we evaluate ControlNet-based approaches, including SD3 ControlNet[4] and FLUX ControlNet[5]. The SD3 ControlNet model is fine-tuned on a large-scale dataset of 12 million image–mask pairs at 1024×1024 resolution. Similarly, FLUX ControlNet is trained on 15 million images.

We also include FLUX Fill, a FLUX-based checkpoint trained specifically for inpainting[6]. For video inpainting, we evaluate Wan2.1-VACE 1.3B, a dedicated video inpainting checkpoint.

Training-free baselines. We adopt the hyperparameters reported in Moufad et al. (2025, Appendix Table 8). As FLAIR (Erbach et al., 2025) was not considered in Moufad et al. (2025), we detail below its implementation and hyperparameters choices. We follow Erbach et al. (2025, Algorithm 1) and build upon their official codebase[7]. We set the number of inner likelihood optimization steps to `n_likelihood_steps`$= 15$ with early stopping threshold `early_stopping`$= 10^{-4}$. Since we consider different prior models from those in the original work, namely SD3, we use a fixed regularization weight of 1, which we found to perform robustly in practice without requiring calibration from the pre-computed flow-matching loss.

B Inpainting in latent space

A key observation of Avrahami et al. (2023) is that, for the case of inpainting, guidance can be carried out entirely in latent space by deriving an appropriate latent mask: the binary pixel space mask is downsampled according; while The input is directly encoded as $x_* = \mathcal{E}(\text{input})$. This equivalence allows inpainting problems to be formulated directly in latent space and benefit from the its compressed representations. Moreover, it enables the application of training-free guidance methods that are valid only for linear inverse problems; in particular, the closed-form update in Equation (2.5) applies in this setting.

Limitations. Defining inpainting directly in latent space entails several limitations, as noted by Avrahami et al. (2023). First, the quality of the reconstruction is inherently constrained by the encoder–decoder pair of the diffusion model, although this limitation becomes less pronounced as recent models continue to improve (Labs, 2025). Second, the encoder's downsampling factor limits the spatial granularity of masks that can be represented: masked regions smaller than the downsampling scale may be ignored, leading to ineffective or inaccurate edits. Additionally, small or thin masked regions can introduce artifacts due to leakage from unmasked regions, since the downsampled mask

[3] https://github.com/Badr-MOUFAD/ding-editor

[4] huggingface.co/alimama-creative/SD3-Controlnet-Inpainting

[5] huggingface.co/alimama-creative/FLUX.1-dev-Controlnet-Inpainting-Beta

[6] huggingface.co/black-forest-labs/FLUX.1-Fill-dev

[7] https://github.com/prs-eth/FLAIR/

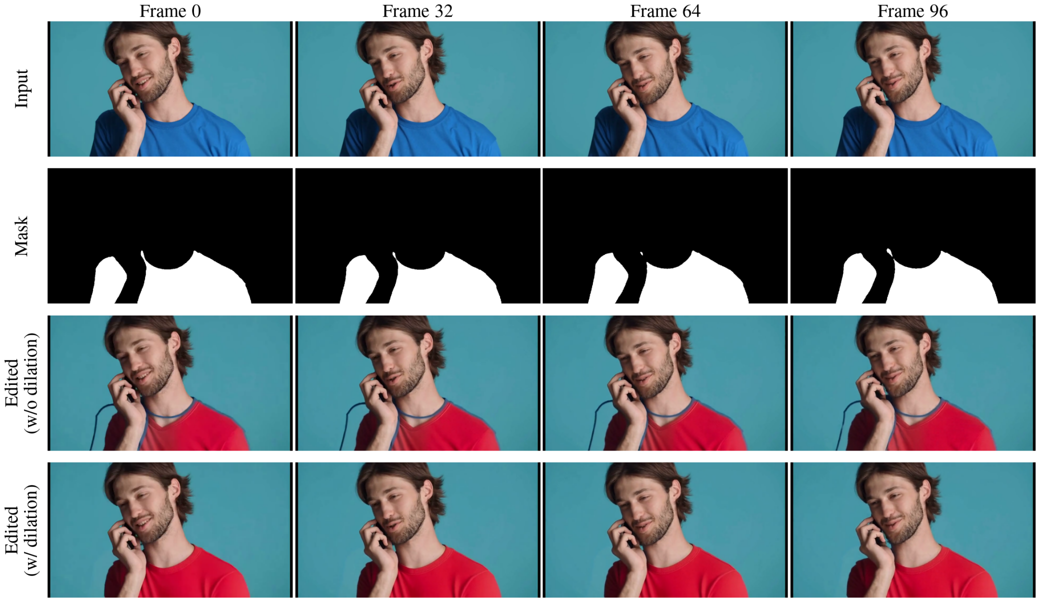


Figure 3: Visualization of *context leakage* in latent-space video inpainting. When lifting a pixel-space inpainting task to the latent space, downsampling the mask without adjustment can lead to boundary artifacts. From top to bottom row: input video, binary edit masks, reconstruction using naive downsampled masks, and reconstruction using the dilated masks. Note that naive downsampling (third row) causes the t-shirt's original boundary (blue outline) to leak into the latent reconstruction, whereas dilation (fourth row) successfully avoid this issue.

may not fully cover the intended area. Figure 3 illustrates this limitation in practice: the outline of the blue shirt in the input video leaks into the edited result. The effect becomes more pronounced as the encoder's downsampling factor increases. In particular, we noticed in practice that this issue is more evident for the LTX video model, which employs a spatial downsampling factor of ×32, compared to Wan2.1, which uses a factor of ×8. We mitigated this issue by slightly overestimating the support of the mask near the boundaries as illustrated in Figure 1, which alleviates boundary artifacts at the cost of slightly altering the observed regions around the mask edges.

C Examples of editing via inpainting

Here, we provide a side-by-side comparison of the DInG and the considered baselines on image editing tasks via inpainting on `HumanEdit`. The red overlay in the first column shows the masked region to be edited and the text in the left-hand side of each row represents the editing prompt.

Qualitative results on video editing tasks can be found on the project webpage[8].

(See the next pages for the gallery of examples)

[8] https://badr-moufad.github.io/ding-editor/

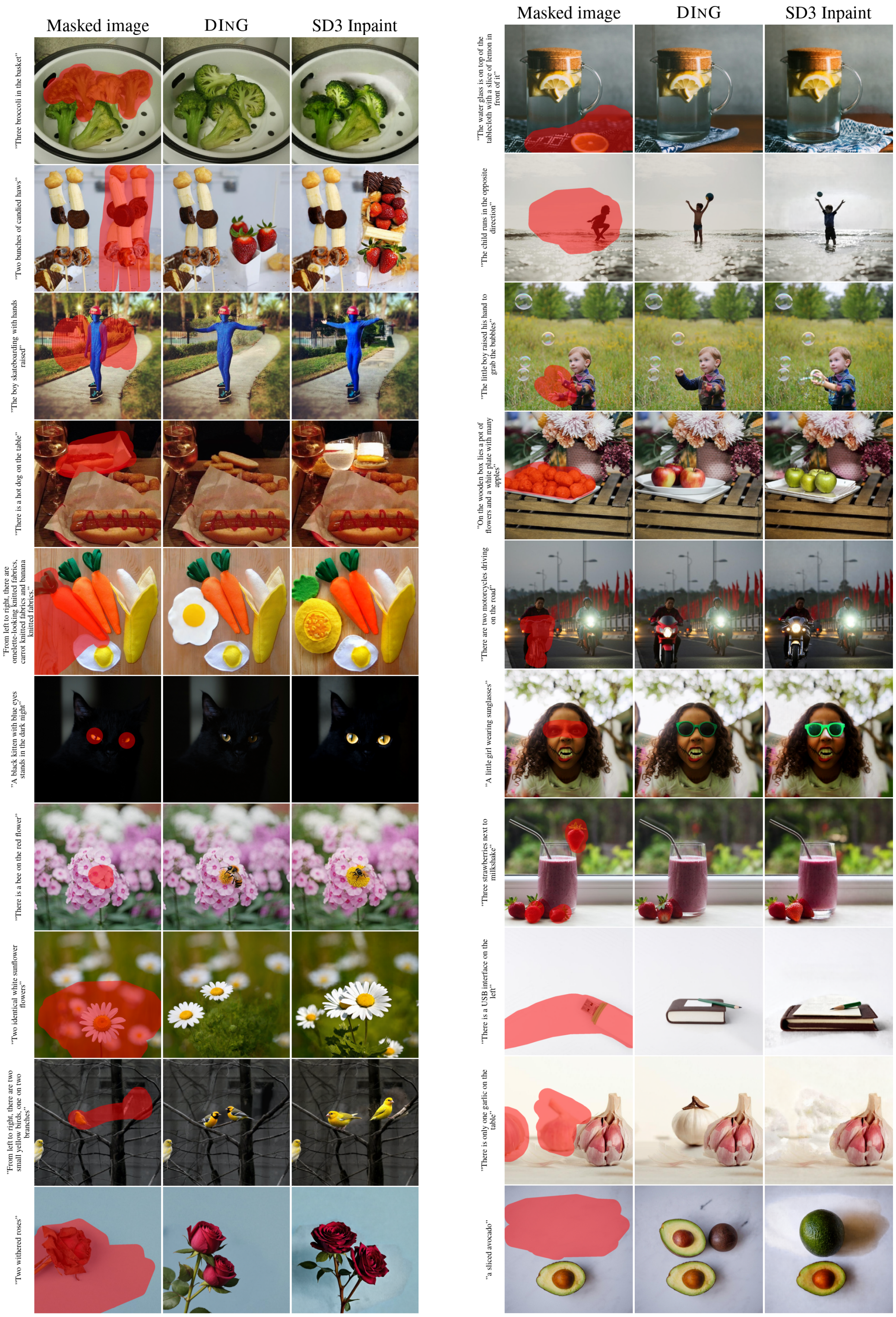

Masked image
DING
SD3 Inpaint
"Three broccoli in the basket"
"Two bunches of candied haws"
"The boy skateboarding with hands raised"
"There is a hot dog on the table"
"From left to right, there are omelette-looking knitted fabrics, carrot knitted fabrics and banana knitted fabrics."
"A black kitten with blue eyes stands in the dark night"
"There is a bee on the red flower"
"Two identical white sunflower flowers"
"From left to right, there are two small yellow birds, one on two branches"
"Two withered roses"
Masked image
DING
SD3 Inpaint
"The water glass is on top of the tablecloth with a slice of lemon in front of it"
"The child runs in the opposite direction"
"The little boy raised his hand to grab the bubbles"
"On the wooden box lies a pot of flowers and a white plate with many apples"
"There are two motorcycles driving on the road"
"A little girl wearing sunglasses"
"Three strawberries next to milkshake"
"There is a USB interface on the left"
"There is only one garlic on the table"
"a sliced avocado"

Figure 4: Qualitative comparison of DING and SD3 with ControlNet (SD3 Inpaint) on `HumanEdit`. The methods are limited to a runtime of 30 seconds per image

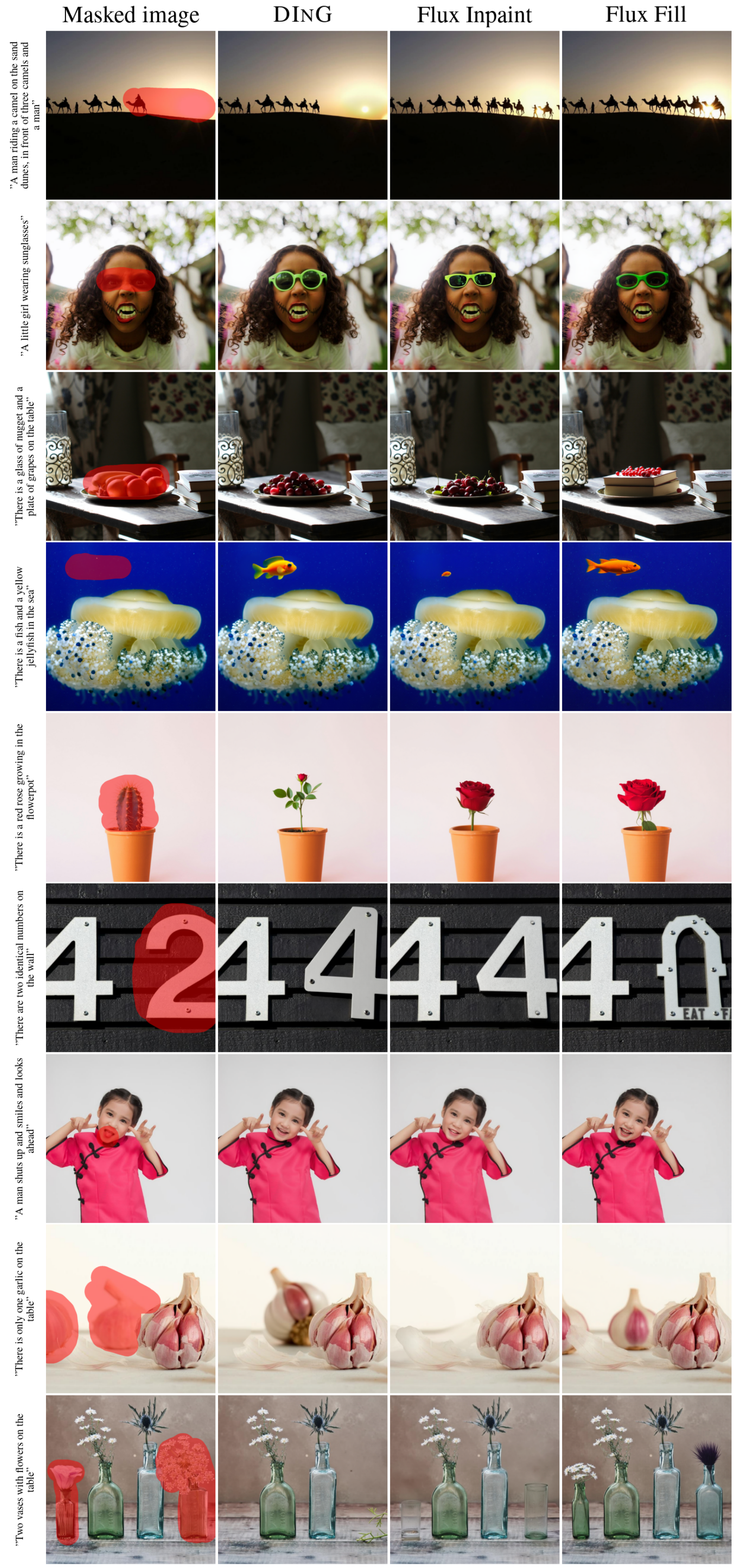

Masked image
DING
Flux Inpaint
Flux Fill
"A man riding a camel on the sand dunes, in front of three camels and a man"
"A little girl wearing sunglasses"
"There is a glass of nugget and a plate of grapes on the table"
"There is a fish and a yellow jellyfish in the sea"
"There is a red rose growing in the flowerpot"
"There are two identical numbers on the wall"
"A man shuts up and smiles and looks ahead"
"There is only one garlic on the table"
"Two vases with flowers on the table"

Figure 5: Qualitative comparison 1 of DING Flux with ControlNet (Flux Inpaint), and Flux Fill on `HumanEdit`. The methods are limited to a runtime of 30 seconds per image.

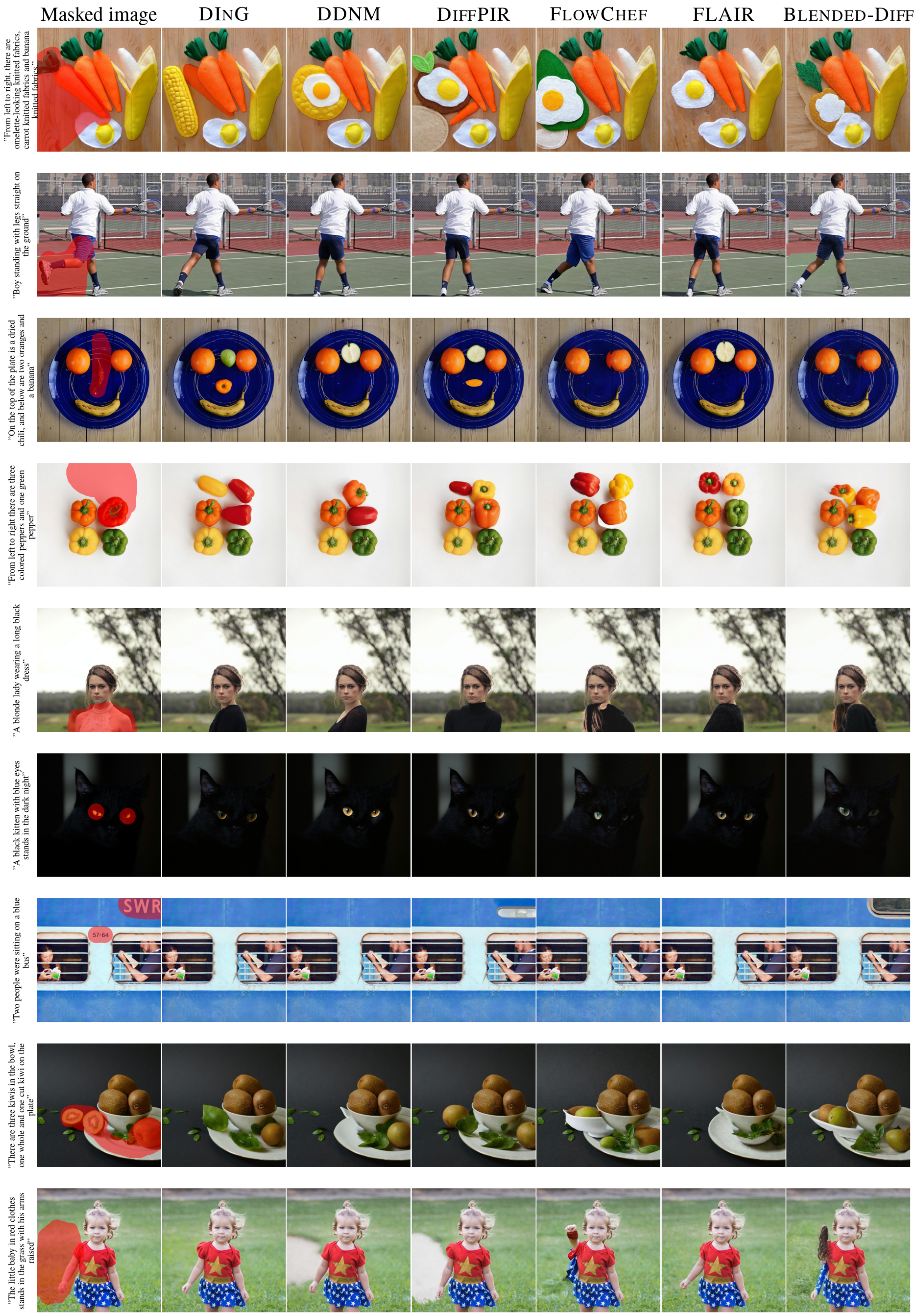

Masked image
DING
DDNM
DIFFPIR
FLOWCHEF
FLAIR
BLENDED-DIFF
"From left to right, there are omelette-looking knitted fabrics, carrot knitted fabrics and banana knitted fabrics."
"Boy standing with legs straight on the ground"
"On the top of the plate is a dried chili, and below are two oranges and a banana"
"From left to right there are three colored peppers and one green pepper"
"A blonde lady wearing a long black dress"
"A black kitten with blue eyes stands in the dark night"
"Two people were sitting on a blue bus"
"There are three kiwis in the bowl, one whole and one cut kiwi on the plate"
"The little baby in red clothes stands in the grass with his arms raised"
SWR
57-64

Figure 6: Qualitative comparison between and training-free baselines on `HumanEdit` with SD3 model as prior. The methods are limited to NFE 50.