ON-POLICY SELF-DISTILLATION FOR MULTI-TURN IMAGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-based image editing has achieved strong performance in single-turn settings, yet practical editing is often iterative, with each instruction applied to the output of the previous turn. We find that existing editing models degrade rapidly under recursive editing and attribute this failure to a train–test mismatch in the conditioning distribution: models are trained on clean source images but must repeatedly condition on their own imperfect outputs at inference time. To address this, we propose MT-OPSD, an on-policy self-distillation framework that trains the model on self-generated conditioning states with editing supervision from a clean-conditioned teacher, without requiring multi-turn annotations. We further introduce LME-Bench, a benchmark of 100 ten-turn editing sessions for evaluating long-horizon robustness. Experiments across three editing backbones show that MT-OPSD substantially improves long-horizon editing success and reduces multi-turn collapse while largely preserving single-turn editing quality. All code, checkpoints, and data will be publicly released.

1 INTRODUCTION

Recent progress in instruction-based image editing (Brooks et al., 2023; Wei et al., 2025; Wu et al., 2025a) has made it possible to modify images through natural-language instructions, covering a broad range of operations from local object edits to global appearance changes. Despite strong performance on single-turn edits, real-world editing workflows often involve multiple rounds of refinement. For example, a user may adjust the lighting of a scene, modify its color palette, reposition an object, and apply a stylistic transformation, with each edit operating on the result of the previous one. A practical image editor should therefore remain reliable across multiple editing turns, maintaining visual coherence while accurately following each new instruction.

In practice, however, current editing models often struggle across consecutive editing turns. As a model repeatedly edits its own outputs, small errors accumulate and progressively degrade image quality. After only a few turns, outputs may exhibit high-frequency chromatic noise, structural fragmentation, or severe identity drift. We observe this behavior across different editing models, suggesting that multi-turn degradation is a broader limitation of the standard single-turn training paradigm rather than a model-specific failure.

We hypothesize that this degradation arises from a train–test mismatch in the conditioning distribution. During training, editing models are conditioned on clean source images, whereas at inference time they must operate on their previous outputs, which inevitably contain small model-induced errors. Although these errors may be visually negligible after a single edit, they accumulate as each output becomes the input to the next turn. Figure 1 provides a simple diagnostic: even when instructed to reproduce the input image without modification, the model exhibits measurable drift at each turn, indicating that recursive editing in-

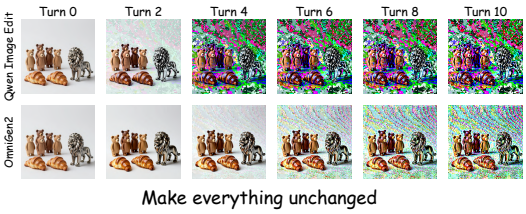

Figure 1: **Drift under identity editing.** Repeatedly instructing a model to leave the image unchanged exposes model-induced errors that accumulate across turns, and similar degradation occurs across different editing models.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

troduces errors that are not captured by single-turn evaluation. This behavior is analogous to exposure bias (Huang et al., 2026b) in autoregressive video generation, where models trained on ground-truth data are evaluated on their own outputs.

One possible solution is to train directly on multi-turn editing sequences, but such data is scarce. More importantly, optimizing end-to-end through model-generated multi-turn rollouts would require backpropagation across hundreds of denoising steps and multiple editing turns, making it prohibitively expensive. Training-free methods such as Emu Edit (Sheynin et al., 2024) provide another option by reverting nearly unchanged pixels to each turn's input, which can reduce drift for local edits. However, this approach does not extend to global transformations such as restyling or relighting, where most pixels are expected to change.

To address this gap, we propose MT-OPSD, an on-policy self-distillation framework for robust multi-turn image editing without requiring multi-turn annotations or ground-truth edited images. MT-OPSD builds on the observation that a pretrained editor already exhibits strong single-turn editing behavior under clean conditioning; multi-turn editing introduces an additional challenge as model-induced errors are repeatedly carried into subsequent turns. Rather than modeling the full distribution of multi-turn editing histories, we isolate this error component through identity rollouts, which preserve the intended image content while accumulating errors from the model's own predictions. Training then alternates between two complementary objectives: an identity branch that prevents further drift, and an editing branch that pairs these self-generated states with real editing instructions and transfers the model's clean-condition editing behavior through on-policy velocity matching. An adaptive rollout curriculum progressively exposes the student to deeper self-generated states, while gated teacher promotion updates the clean-condition reference as training proceeds.

To facilitate the evaluation of long-horizon editing robustness, we further construct Long Multi-turn Image Editing Bench (LME-Bench), an evaluation benchmark consisting of 100 editing sessions, each containing 10 consecutive turns with a diverse combination of local and global operations. Each session is evaluated at every turn in terms of editing accuracy, visual consistency, and image quality, enabling a systematic analysis of when and how multi-turn degradation emerges.

In summary, our contributions are as follows:

- We show that multi-turn collapse occurs across modern image editing models and attribute it to the train–test mismatch in the conditioning distribution.
- We propose MT-OPSD, an on-policy self-distillation framework that transfers the model's own clean-conditioned editing behavior to self-generated states, without requiring multi-turn annotations or an external teacher.
- We introduce LME-Bench, a benchmark of 100 ten-turn editing sessions covering both local and global operations for evaluating long-horizon editing robustness.
- Experiments across three editing backbones and multiple benchmarks show that MT-OPSD substantially improves long-horizon editing success and reduces multi-turn collapse while largely preserving single-turn editing quality.

2 Related Work

Instruction-based Image Editing. The field of image editing has witnessed a paradigm shift from domain-specific generative adversarial networks (Goodfellow et al., 2020) to high-fidelity diffusion models (Ho et al., 2020; Rombach et al., 2022; Abdelrahman et al., 2025). Early diffusion-based approaches (Hertz et al., 2022; Mokady et al., 2023; Zhao et al., 2023) explored attention manipulation and latent inversion to enable image modifications while preserving the original content, but often struggled with complex and diverse editing instructions. This motivated instruction-based editing, pioneered by InstructPix2Pix (Brooks et al., 2023) and subsequently advanced through improved data curation and scaling (Zhang et al., 2023; Wei et al., 2025; Zhuo et al., 2025), stronger vision-language understanding (Wu et al., 2025a;b; Zhao et al., 2026a), unified generative frameworks (Deng et al., 2025; Xie et al., 2024), and in-context flow models (Labs et al., 2025; Liu et al., 2025). However, these approaches are primarily designed and evaluated for independent single-turn edits, leaving the robustness of models under repeated self-conditioned editing largely unexplored.

Multi-turn Image Editing. Recent works have explored improving consistency across sequential edits. Training-free approaches, including Emu Edit (Sheynin et al., 2024), FreqEdit (Liao et al., 2026), and VAE-LFA (Wang et al., 2026), alleviate accumulated degradation through image-space or latent-space corrections, but rely on assumptions about the edits and do not alter the model's editing behavior. MTC (Zhou et al., 2025) uses per-turn inversion with trajectory control and adaptive attention guidance on text-to-image models. VINCIE (Qu et al., 2026) and AnchorEdit (Xu et al., 2026) instead train dedicated models for causal multi-turn editing by adapting video architectures to interleaved image sequences. Edit-R2 (Ye et al., 2026b) focuses on preserving session-level constraints across turns. Most closely related to our work, MT-EditFlow (Huang et al., 2026a) also attributes multi-turn degradation to exposure bias, but addresses it through reinforcement learning with external reward supervision. In contrast, MT-OPSD starts from the observation that pretrained editors already possess strong single-turn editing ability, and uses the model's own clean-condition behavior to extend this ability to self-generated states through on-policy self-distillation.

On-Policy Self-Distillation. On-policy self-distillation (OPSD) trains a model on states generated by its own policy while using the same model as teacher under richer conditioning. In language models, this is commonly realized by providing the teacher with privileged context (Zhao et al., 2026b; Penaloza et al., 2026; Sang et al., 2026). In visual generation, D-OPSD (Jiang et al., 2026) conditions the teacher on a paired target image while supervising the student along its own diffusion trajectory, whereas OPSD-V (Liu et al., 2026) uses real long-video context to supervise autoregressive generation from self-generated history. DiffusionOPSD (Zhou et al., 2026a) derives its targets from reward gradients rather than privileged context. Closely related OPD methods for diffusion and flow models likewise provide teacher supervision along student-generated sampling trajectories (Li et al., 2026; Fang et al., 2026; Zhou et al., 2026b). Our setting introduces a different form of on-policy state: in multi-turn editing, the conditioning image itself evolves through the model's previous outputs. MT-OPSD therefore uses the clean source image as privileged teacher context and the self-generated rollout state as student context, extending the model's clean-condition editing behavior to recursive editing without paired target images or multi-turn annotations.

3 METHOD

3.1 PRELIMINARIES AND PROBLEM FORMULATION

Multi-turn Image Editing. Modern instruction-based editing models are commonly built on flow matching (Lipman et al., 2023), where a velocity model $v_\theta(x_t, t, I, e)$ learns a time-dependent vector field between the target edit and Gaussian noise, conditioned on a source image I and an editing instruction e. Given an initial image $I^{(0)}$ and instructions $e_1, \ldots, e_K$, multi-turn editing recursively applies the editing operator $G_\theta(I, e)$:

$$I^{(k)} = G_\theta(I^{(k-1)}, e_k). \tag{1}$$

This multi-turn process introduces a train–test mismatch in the conditioning distribution. While training exposes the model only to clean source images, later turns condition on self-generated outputs $I^{(k-1)}$. Such states contain both the intended semantic changes introduced by earlier edits and model-induced errors accumulated across previous editing turns. The former are part of the editing task itself, while the latter are absent from clean single-turn training and constitute the additional source of mismatch that we target.

On-Policy Self-Distillation. On-policy self-distillation (OPSD) combines on-policy distillation (Agarwal et al., 2024) with self-distillation: the student is supervised at states generated by its own policy, while the same model serves as the teacher under a more informative context (Zhao et al., 2026b; Penaloza et al., 2026). Let s denote a state visited along the student rollout, and let c_S and c_T denote the student and teacher contexts, respectively. For a student π_θ and a teacher $\pi_{\bar{\theta}}$ derived from the same model, OPSD minimizes

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{s \sim d_{\pi_\theta}(\cdot \mid c_S)} \left[D(\pi_{\bar{\theta}}(\cdot \mid s, c_T), \pi_\theta(\cdot \mid s, c_S)) \right], \tag{2}$$

where d_{π_θ} denotes the state distribution induced by the student and D is a distillation divergence. This allows the model to distill information available under a richer context into its behavior under a

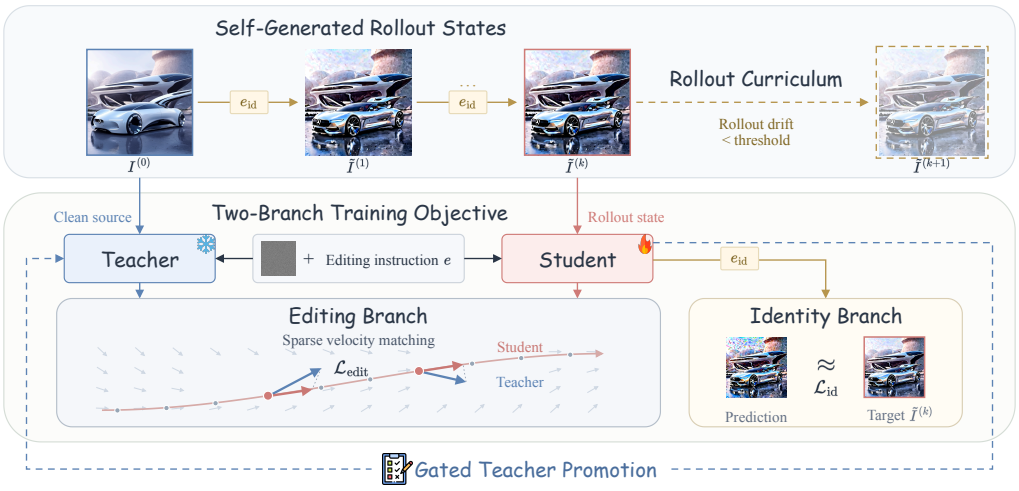

Figure 2: **Overview of MT-OPSD.** The student first constructs self-generated rollout states by repeatedly applying the identity instruction. For each rollout state $\tilde{I}^{(k)}$, the identity branch uses the state itself as the target under $\mathcal{L}_{\text{id}}$, while the editing branch aligns the student conditioned on $\tilde{I}^{(k)}$ with a frozen teacher conditioned on the clean source $I^{(0)}$ via sparse query-based velocity matching $\mathcal{L}_{\text{edit}}$. A rollout curriculum progressively increases the rollout depth, and improved student checkpoints replace the teacher through gated promotion.

less informative student context, without a separate teacher. For diffusion models, OPSD is typically instantiated by constructing asymmetric conditioning contexts for the teacher and student, so that predictions under the teacher context provide supervision along the student's denoising process.

3.2 MT-OPSD

To address the conditioning mismatch, MT-OPSD isolates the model-induced error component and uses on-policy self-distillation to preserve the model's clean-condition editing behavior on self-generated states. We denote the student by θ_S and the teacher by θ_T, both initialized from the same pretrained model. The student operates on self-generated states, while the teacher is conditioned on the clean source and remains fixed between gated promotions. The framework consists of four components: self-generated rollout states, a two-branch training objective, an adaptive rollout curriculum, and gated teacher promotion. The overall framework and training procedure of MT-OPSD are summarized in Figure 2 and Algorithm 1.

Self-Generated Rollout States. To obtain conditioning states that contain errors induced by the model itself, we recursively apply the current student to its own outputs. Specifically, we define an identity instruction e_{id} (e.g., "Make everything unchanged") that asks the model to reproduce the input image without modification, and roll out the current student for k turns starting from a clean source image $I^{(0)}$:

$$\tilde{I}^{(k)} = G_{\theta_S}(\tilde{I}^{(k-1)}, e_{\text{id}}), \qquad \tilde{I}^{(0)} = I^{(0)}. \tag{3}$$

The rollout follows the same sampling configuration as inference, so the resulting drift arises from the model's own generation process. Since the intended image content remains unchanged, the difference between $\tilde{I}^{(k)}$ and $I^{(0)}$ mainly reflects model-induced errors accumulated across the rollout. Using actual editing instructions during the rollout would instead entangle these errors with intended semantic changes, making it difficult to obtain a reliable supervision signal for learning robustness to model-induced errors. The identity rollout therefore provides a controlled way to construct on-policy conditioning states from single-turn training data while isolating the error component.

Two-Branch Training Objective. Given a rollout state $\tilde{I}^{(k)}$, we optimize two complementary branches, sampled at a fixed ratio. The identity branch operates under e_{id} and suppresses unintended

changes and accumulated errors across turns, while the editing branch pairs the same rollout state with a real editing instruction e to preserve editing capability under self-generated conditioning.

The identity branch supervises the model under the identity instruction. A natural choice is to use the clean source image $I^{(0)}$ as the target, asking the model to recover the clean image from its degraded input. However, we find this restoration objective difficult to optimize, as it requires correcting errors accumulated over k turns within a single turn. We therefore use the rollout state itself as the target, yielding an identity objective that prevents further drift rather than restoring the clean source:

$$\mathcal{L}_{\text{id}} = \mathbb{E}_{t,\epsilon}\left[\left\| v_{\theta_S}(\tilde{x}_t, t, \tilde{I}^{(k)}, e_{\text{id}}) - (\epsilon - \tilde{x}_0)\right\|_2^2\right], \tag{4}$$

where $\tilde{x}_0$ denotes the latent representation of $\tilde{I}^{(k)}$, $\tilde{x}_t = (1-t)\tilde{x}_0 + t\epsilon$, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. This objective prevents accumulated errors from being further amplified, but provides no supervision for executing non-identity edits on degraded states.

The editing branch addresses this limitation by maintaining the model's editing capability under self-generated conditioning. Because the identity rollout preserves the intended content of $I^{(0)}$, the rollout state $\tilde{I}^{(k)}$ and the clean source differ primarily in accumulated model-induced errors. For the same editing instruction e, we therefore use the model's clean-conditioned prediction as a reference for editing $\tilde{I}^{(k)}$. Specifically, the teacher θ_T is conditioned on $I^{(0)}$, while the student θ_S is conditioned on $\tilde{I}^{(k)}$. Following the sparse query-based velocity matching of DanceOPD (Zhou et al., 2026b), the student performs an N-step denoising process. We sample a small number of query steps from p_q and match the teacher and student velocities at the corresponding student states:

$$\mathcal{L}_{\text{edit}} = \mathbb{E}_{q\sim p_q}\left[\left\| v_{\theta_S}(\bar{x}_{t_q}, t_q, \tilde{I}^{(k)}, e) - v_{\theta_T}(\bar{x}_{t_q}, t_q, I^{(0)}, e)\right\|_2^2\right], \tag{5}$$

where $\bar{x}_{t_q} = \text{sg}(x_{t_q})$ denotes the stop-gradient state reached by the student at query step q. At each queried step, teacher and student are evaluated at the same noisy latent with the same classifier-free guidance (CFG) scale, and each retains its own image condition in both CFG branches, dropping only the text condition in the unconditional branch.

Rollout Curriculum. The difficulty of both training branches increases with rollout depth k, as deeper rollouts accumulate larger model-induced errors. Starting from a large depth therefore exposes the model to heavily degraded states early in training and can destabilize optimization. We instead increase the rollout depth progressively. Specifically, we measure rollout drift as the mean pixel deviation between $\tilde{I}^{(k)}$ and $I^{(0)}$, and increase the depth by one turn only when the drift remains below a threshold for a fixed number of consecutive training steps. After each increase, the resulting rise in drift delays further progression until the model adapts to the current depth, yielding an adaptive curriculum without a manually specified schedule. We cap the rollout depth at $K_{\max}$. In practice, the adaptive curriculum typically saturates at around four turns, well below this limit.

Gated Teacher Promotion. The editing branch relies on θ_T as a stable clean-condition reference. The pretrained initialization provides a strong starting teacher because it already exhibits reliable single-turn editing behavior under clean conditioning. As the student is trained on self-generated states with the two-branch objective, however, it can gradually acquire greater robustness to self-generated conditioning than the current teacher. Continuing to match the same fixed teacher can then limit further improvement. We therefore allow improved student checkpoints to replace the teacher, while keeping the teacher fixed between promotions. Each candidate is evaluated asynchronously by a VLM judge on a held-out gate set and promoted only when it satisfies a long-horizon criterion based on success and collapse rates. The VLM judge is used only for checkpoint selection rather than as a training target or reward signal. The full promotion rule is provided in Appendix C.

4 Experiments

4.1 LME-Bench

Benchmark Construction. Existing multi-turn editing benchmarks contain at most five consecutive turns, making them insufficient to evaluate the long-horizon degradation studied in this work.

We therefore construct LME-Bench, which contains 100 editing sessions of 10 consecutive turns each, for a total of 1,000 editing instructions. Source images are generated by Z-Image-Turbo (Cai et al., 2025) at 1024×1024 resolution and are evenly distributed across 10 semantic categories. As shown in Figure 3, each session contains six local edits that modify specific objects or attributes and four global edits that change the overall style, atmosphere, or photometric appearance. Global edits are placed non-adjacently and are not used in the final turns, so that later local edits are applied to images that have already undergone global changes. To avoid invalid or ambiguous instructions, we define six families of validity rules, covering cases such as editing invisible regions or requesting a state that is already satisfied. Candidate instructions are first checked by a VLM and then manually reviewed to ensure that each edit is visible, feasible, and unambiguous. The full edit taxonomy and composition rules are provided in Appendix D.1.

Evaluation Protocol and Metrics. Each session is executed sequentially, with every turn conditioned only on the output of the previous turn. We use GPT-4o to evaluate each turn in terms of prompt following, consistency with the previous state, and visual quality. We report two complementary metrics. Success rate at turn k (SR@k) is the fraction of sessions in which all of the first k turns satisfy both the prompt-following and consistency criteria, reflecting cumulative editing success. Collapse rate at turn k (CR@k) is the fraction of sessions that have undergone persistent visual degradation by turn k. A session is considered collapsed once two consecutive turns are judged visually degraded, and remains counted as collapsed thereafter. The full judging prompts, thresholds, and evaluation details are provided in Appendix D.2.

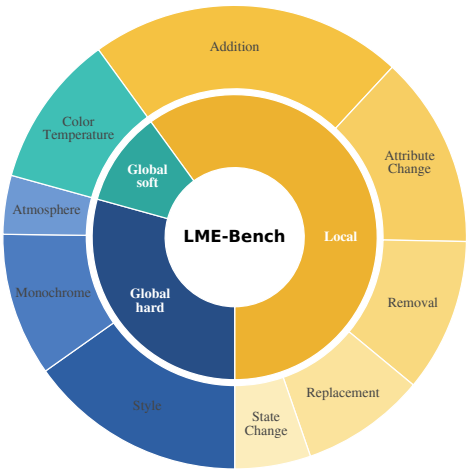

Figure 3: Edit-type distribution in LME-Bench.

4.2 Experimental Setup

Implementation Details. We instantiate MT-OPSD on three instruction-based editing backbones, including Qwen-Image-Edit-2511 (Wu et al., 2025a), FLUX.2-klein-base (Labs, 2025), and FireRed-Image-Edit (Team et al., 2026), and adapt each model using LoRA (Hu et al., 2022). Unless otherwise specified, we use a LoRA rank of 32, set the maximum rollout depth to $K_{\max} = 10$, and sample the editing and identity branches at a ratio of 2:1. Training uses approximately 2,000 source-image/instruction pairs from OmniEdit (Wei et al., 2025); the corresponding ground-truth edited images are not used. All self-rollouts follow the same sampling configuration as inference, and GPT-4o is used only for gated teacher promotion during training. Each run uses four NVIDIA H100/H200 GPUs for training and another four GPUs for asynchronous gate evaluation. Additional implementation details and hyperparameters are provided in Appendix A.

Evaluation. We evaluate long-horizon editing robustness on LME-Bench, using SR@k and CR@k as defined in Section 4.1. We also evaluate on MSE-Bench (Qu et al., 2026), which contains 100 five-turn editing sessions, with each turn applied to the output of the previous one. Following its official evaluation protocol, GPT-4o scores each turn for prompt following and consistency with the previous state, and a turn is considered successful only when both scores meet the corresponding thresholds. We report the success rate at each editing turn. To verify that the improved multi-turn robustness does not sacrifice single-turn editing quality, we further include ImgEdit (Ye et al., 2026a), a standard single-turn instruction-based editing benchmark, and report its overall score.

Comparison Baselines. For each backbone, we compare MT-OPSD against the original base model and three training-free methods for reducing multi-turn degradation. Emu Edit (Sheynin et al., 2024) reverts nearly unchanged pixels to each turn's input. FreqEdit (Liao et al., 2026) mitigates accumulated frequency degradation by recovering high-frequency information, while VAE-

Table 1: **Quantitative comparison on LME-Bench.** We report the success rate (SR) and the collapse rate (CR) at turns 3, 5, 8, and 10.

Models	SR ↑				CR ↓			
	Turn-3	Turn-5	Turn-8	Turn-10	Turn-3	Turn-5	Turn-8	Turn-10
Proprietary Models								
GPT-Image-1 (OpenAI, 2025)	0.82	0.48	0.09	0.01	0.03	0.35	0.63	0.69
GPT-Image-2 (OpenAI, 2026)	1.00	0.98	0.98	0.96	0.00	0.00	0.00	0.00
Nano Banana (Google, 2025a)	0.67	0.66	0.63	0.56	0.00	0.00	0.00	0.00
Nano Banana Pro (Google, 2025b)	1.00	1.00	0.97	0.91	0.00	0.00	0.00	0.00
Open-Source Models								
Qwen-Image-Edit-2511 (Wu et al., 2025a)	0.98	0.43	0.08	0.03	0.00	0.10	0.45	0.55
+ Emu Edit (Sheynin et al., 2024)	0.95	0.54	0.19	0.13	0.00	0.04	0.21	0.30
+ FreqEdit (Liao et al., 2026)	0.89	0.22	0.03	0.01	0.00	0.08	0.41	0.54
+ VAE-LFA (Wang et al., 2026)	0.84	0.46	0.18	0.10	0.00	0.05	0.29	0.36
+ MT-OPSD	**1.00**	**0.91**	**0.63**	**0.44**	**0.00**	**0.01**	**0.02**	**0.02**
FireRed-Image-Edit (Team et al., 2026)	0.95	0.57	0.28	0.15	0.01	0.16	0.51	0.61
+ Emu Edit (Sheynin et al., 2024)	0.97	0.58	0.30	0.14	0.01	0.05	0.36	0.51
+ FreqEdit (Liao et al., 2026)	0.93	0.41	0.04	0.00	0.01	0.11	0.64	0.73
+ VAE-LFA (Wang et al., 2026)	0.88	0.70	0.50	0.37	0.01	0.08	0.26	0.33
+ MT-OPSD	**0.98**	**0.88**	**0.64**	**0.52**	**0.00**	**0.00**	**0.01**	**0.03**
FLUX.2-klein-base (Labs, 2025)	0.92	0.52	0.29	0.12	0.00	0.01	0.17	0.25
+ Emu Edit (Sheynin et al., 2024)	0.90	0.52	0.23	0.11	0.00	0.02	0.12	0.18
+ FreqEdit (Liao et al., 2026)	0.64	0.19	0.03	0.01	0.00	0.03	0.35	0.54
+ VAE-LFA (Wang et al., 2026)	0.81	0.51	0.31	0.16	0.00	0.03	0.16	0.24
+ MT-OPSD	**0.95**	**0.76**	**0.57**	**0.38**	**0.00**	**0.00**	**0.02**	**0.04**

LFA (Wang et al., 2026) corrects low-frequency drift in the VAE latent space. All methods use the same backbone and inference configuration whenever applicable. We additionally report GPT-Image-1 (OpenAI, 2025) and Nano Banana (Google, 2025a) as proprietary systems, together with the more recent GPT-Image-2 (OpenAI, 2026) and Nano Banana Pro (Google, 2025b) on LME-Bench. Among the training-based multi-turn editing methods discussed in Section 2, VINCIE (Qu et al., 2026) is the only one with public code and checkpoints. Since VINCIE is trained as a separate model rather than instantiated on our backbones, we report its comparison separately in Appendix E.

4.3 Main Results

Results on LME-Bench. Table 1 shows that all three open-source base models degrade sharply over longer editing sequences, with SR@10 falling to 0.03–0.15 and CR@10 rising to 0.25–0.61. Existing training-free methods yield model-dependent gains: Emu Edit notably improves Qwen-Image-Edit-2511, VAE-LFA is more effective on FireRed-Image-Edit, while FreqEdit generally degrades long-horizon performance. In contrast, MT-OPSD raises SR@10 to 0.38–0.52 across all three backbones and reduces CR@10 to at most 0.04 without sacrificing early-turn performance. Although the rollout curriculum typically saturates at around four turns, these gains persist through ten-turn evaluation, indicating robustness beyond the rollout depths encountered during training. Among proprietary systems, GPT-Image-1 also exhibits severe long-horizon degradation and is outperformed by MT-OPSD across all reported turns and metrics. Nano Banana rarely collapses, while its lower SR is primarily attributable to failures on global edits that occur predominantly in the earlier turns. GPT-Image-2 and Nano Banana Pro remain stronger at long horizons. Overall, these results show that robustness learned from self-generated rollout states transfers to real multi-turn editing sequences, substantially narrowing the long-horizon robustness gap for open-source models.

Results on MSE-Bench and ImgEdit. Table 2 reports results on MSE-Bench, an existing five-turn benchmark dominated by local edits, and ImgEdit, which evaluates whether the improved multi-turn robustness preserves single-turn editing quality. On MSE-Bench, MT-OPSD improves SR@5 from 0.24 to 0.51 on Qwen-Image-Edit-2511 and from 0.40 to 0.69 on FireRed-Image-Edit, while matching FLUX.2-klein-base at 0.49. Emu Edit remains competitive in this setting, where the predominance of local edits favors pixel-level restoration, but MT-OPSD achieves comparable or better performance across the three backbones. On ImgEdit, MT-OPSD largely preserves the original single-turn capability, with only minor score changes across all three models.

Table 2: **Quantitative comparison on MSE-Bench and ImgEdit.** We report the success rate (SR) at each turn on MSE-Bench under its official evaluation protocol, together with the overall score on the single-turn ImgEdit benchmark.

Models	MSE-Bench (SR ↑)					ImgEdit Overall ↑
	Turn-1	Turn-2	Turn-3	Turn-4	Turn-5	
Proprietary Models						
GPT-Image-1 (OpenAI, 2025)	0.96	0.69	0.67	0.64	0.56	4.20
Nano Banana (Google, 2025a)	0.99	0.77	0.75	0.73	0.63	4.29
Open-Source Models						
Qwen-Image-Edit-2511 (Wu et al., 2025a)	0.97	0.91	0.72	0.52	0.24	4.51
+ Emu Edit (Sheynin et al., 2024)	**0.99**	**0.93**	**0.80**	0.69	0.49	4.36
+ FreqEdit (Liao et al., 2026)	0.95	0.80	0.60	0.39	0.14	4.48
+ VAE-LFA (Wang et al., 2026)	0.99	0.93	0.78	0.61	0.34	4.32
+ MT-OPSD	0.98	0.90	0.79	**0.69**	**0.51**	4.49
FireRed-Image-Edit (Team et al., 2026)	**1.00**	**0.97**	0.80	0.62	0.40	4.56
+ Emu Edit (Sheynin et al., 2024)	1.00	0.95	0.86	0.77	0.59	4.33
+ FreqEdit (Liao et al., 2026)	0.99	0.91	0.70	0.48	0.24	4.54
+ VAE-LFA (Wang et al., 2026)	0.99	0.95	0.84	0.67	0.48	4.29
+ MT-OPSD	0.99	0.95	**0.87**	**0.78**	**0.69**	4.52
FLUX.2-klein-base (Labs, 2025)	**0.97**	0.85	0.77	0.66	0.49	4.20
+ Emu Edit (Sheynin et al., 2024)	0.93	0.85	0.73	0.58	0.42	4.01
+ FreqEdit (Liao et al., 2026)	0.85	0.57	0.35	0.19	0.11	3.91
+ VAE-LFA (Wang et al., 2026)	0.95	0.86	0.69	0.52	0.41	4.08
+ MT-OPSD	0.95	**0.89**	**0.77**	**0.66**	**0.49**	4.28

Qualitative Results. Figure 4 shows selected turns from one ten-turn editing session across three backbones. The base models produce reasonable results in the early turns, but gradually develop severe visual artifacts as their outputs are repeatedly reused as inputs. The training-free baselines provide limited improvement: Emu Edit and FreqEdit largely follow the same degradation pattern as the base models, while VAE-LFA better preserves image quality but weakens the requested color-temperature edit at Turn-5. In contrast, MT-OPSD remains stable through Turn-10 while following subsequent instructions, showing that long-horizon robustness can be improved without weakening the model's ability to perform later edits. More detailed results are shown in Appendix F.

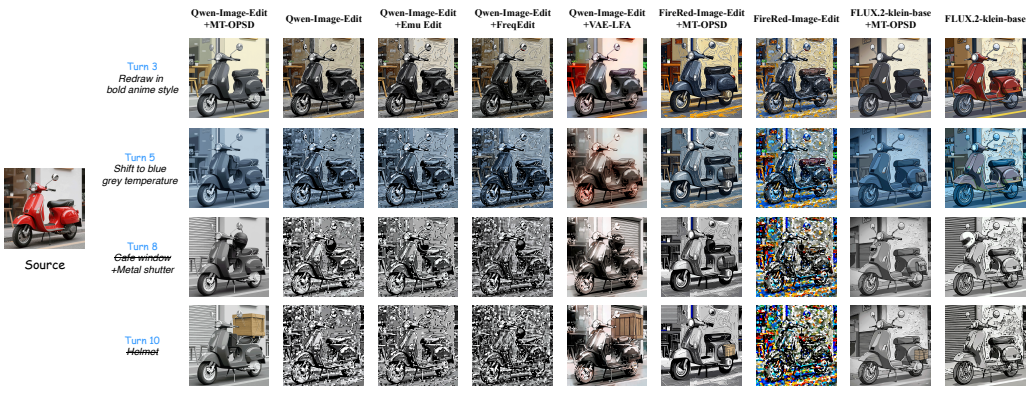

Figure 4: **Qualitative comparison on LME-Bench.** Repeated editing leads to distinct failure patterns across backbones, while the training-free baselines either inherit these artifacts or weaken the requested edits. MT-OPSD remains stable through 10 turns and continues to follow the editing sequence across all three backbones.

Table 3: **Ablation on key components of MT-OPSD.** SFT editing supervision replaces on-policy velocity matching with standard flow-matching supervision toward teacher-generated images. The curriculum ablation fixes the rollout depth at four, where the adaptive curriculum typically saturates.

Method	LME-Bench					MSE-Bench		ImgEdit Overall
	SR@1	SR@5	SR@10	CR@5	CR@10	SR@1	SR@5	
Base model	1.00	0.43	0.03	0.10	0.55	0.97	0.24	4.51
MT-OPSD	**1.00**	**0.91**	0.44	**0.01**	**0.02**	0.98	**0.51**	4.49
w/o identity branch	0.99	0.78	**0.49**	0.03	0.06	0.85	0.50	3.88
w/o editing branch	0.91	0.18	0.00	0.17	0.80	0.95	0.22	4.33
w/ SFT editing supervision	1.00	0.59	0.03	0.12	0.67	0.98	0.37	4.50
w/o rollout curriculum	1.00	0.79	0.17	0.04	0.19	0.96	0.41	4.33
w/o gated promotion	1.00	0.79	0.19	0.01	0.06	**1.00**	0.45	**4.55**

4.4 Ablation Study

All ablations are conducted on Qwen-Image-Edit-2511 under the same training setup. Table 3 reports the quantitative results, with additional visual comparisons provided in Appendix G.

Effect of the Two-Branch Objective. The two training branches serve complementary roles. Removing the editing branch leaves only identity supervision and severely weakens editing ability, with SR@10 falling to 0.00 and CR@10 rising to 0.80. In contrast, removing the identity branch retains strong long-horizon editing performance and even yields a slightly higher SR@10 of 0.49, but makes the edits less controlled: CR@10 increases from 0.02 to 0.06, SR@1 on MSE-Bench drops from 0.98 to 0.85, and the ImgEdit score falls from 4.49 to 3.88. Qualitative results in Appendix G further show that this variant tends to over-edit, progressively altering untargeted content across turns. These results suggest that the editing branch preserves editability on self-generated states, while the identity branch suppresses unnecessary changes and preserves untargeted content.

Effect of On-Policy Supervision. Replacing on-policy velocity matching with standard flow-matching supervision on teacher-generated pseudo-targets reduces SR@5 from 0.91 to 0.59 and SR@10 from 0.44 to 0.03, while CR@10 rises from 0.02 to 0.67, surpassing the base model. Standard flow matching provides supervision along interpolation paths toward teacher-generated targets, whereas sparse velocity matching queries the teacher at states visited by the student. This suggests that supervision on student-visited states is more effective for long-horizon editing.

Effect of the Rollout Curriculum and Gated Promotion. These two components control how the training states and the supervision source evolve during training. Fixing the rollout depth at four turns reduces SR@10 from 0.44 to 0.17 and increases CR@10 from 0.02 to 0.19, showing that progressive depth expansion is more effective than training at a comparable fixed depth. Keeping the pretrained initialization as a fixed teacher instead maintains a low CR@10 of 0.06 but limits SR@10 to 0.19. At the same time, this variant achieves the highest ImgEdit score among the ablations at 4.55, consistent with the view that a fixed clean-condition reference better preserves the pretrained model's single-turn behavior but limits further adaptation to long-horizon editing.

5 Conclusion

In this work, we study the failure of instruction-based editing models under multi-turn editing, where errors in self-generated outputs accumulate into severe degradation. We attribute this failure to a train–test mismatch in the conditioning distribution and introduce MT-OPSD, an on-policy self-distillation framework that trains the model on self-generated conditioning states with editing supervision from a clean-conditioned teacher. We further introduce LME-Bench, a benchmark of 100 ten-turn editing sessions for evaluating long-horizon robustness. Experiments across three editing backbones show that MT-OPSD improves long-horizon robustness on LME-Bench and MSE-Bench while largely preserving single-turn editing quality on ImgEdit. More broadly, our results suggest that strong single-turn editing capability can be extended to multi-turn settings by learning to operate on self-generated states, without multi-turn annotations.

AI USE STATEMENT

We used AI tools to assist with research ideation and hypothesis refinement, experimental design and implementation, result analysis, and manuscript drafting and polishing. Generative models were also part of the research pipeline: they generated synthetic data for LME-Bench and served as judges for benchmark evaluation and checkpoint selection during gated teacher promotion. We manually reviewed AI-assisted outputs and checked experimental code, results, and the final manuscript for accuracy and consistency. We take full responsibility for the content of this work.

ETHICS STATEMENT

Improving the robustness of image editing systems may also make manipulated visual content easier to produce consistently over multiple editing turns. This work does not introduce new editing capabilities, but the resulting models remain subject to the broader risks associated with image manipulation and misuse. All source images in LME-Bench are synthetically generated, and the study involves no human subjects. All datasets and models are used in accordance with their licenses.

REPRODUCIBILITY STATEMENT

We provide the complete training configuration and hyperparameters for all three backbones, the full training algorithm and teacher-promotion rule, and detailed construction and evaluation protocols for LME-Bench, including the judge prompts, in the appendix. All code, model checkpoints, and benchmark data will be publicly released.

REFERENCES

Eslam Abdelrahman, Liangbing Zhao, Tao Hu, Matthieu Cord, Patrick Perez, and Mohamed Elhoseiny. ToddlerDiffusion: Interactive structured image generation with cascaded schrödinger bridge. In *International Conference on Learning Representations*, volume 2025, pp. 65334–65368, 2025.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, 2023.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, et al. Flow-OPD: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Google. Nano Banana, 2025a. URL `https://deepmind.google/models/gemini-image/`.

Google. Nano Banana Pro, 2025b. URL https://deepmind.google/models/gemini-image/pro/.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Jiahui Huang, Yasi Zhang, Tianyu Chen, Shu Wang, Jianwen Xie, Oscar Leong, Mingyuan Zhou, Nanzhu Wang, and Ying Nian Wu. MT-EditFlow: Reinforcement learning for multi-turn image editing with flow matching. *arXiv preprint arXiv:2606.01985*, 2026a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026b.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, et al. D-OPSD: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

Black Forest Labs. FLUX.2: Frontier visual intelligence, 2025. URL https://bfl.ai/blog/flux-2.

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.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. DiffusionOPD: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026.

Yucheng Liao, Jiajun Liang, Kaiqian Cui, Baoquan Zhao, Haoran Xie, Wei Liu, Qing Li, and Xudong Mao. FreqEdit: Preserving high-frequency features for robust multi-turn image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43525–43535, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. OPSD-V: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026.

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.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

OpenAI. GPT-Image-1, 2025. URL https://openai.com/index/image-generation-api/.

OpenAI. GPT-Image-2, 2026. URL https://openai.com/index/introducing-chatgpt-images-2-0/.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Leigang Qu, Feng Cheng, Ziyan Yang, Qi Zhao, Shanchuan Lin, Yichun Shi, Yicong Li, Wenjie Wang, Tat-Seng Chua, and Lu Jiang. VINCIE: Unlocking in-context image editing from video. In *International Conference on Learning Representations*, volume 2026, pp. 114873–114918, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *arXiv e-prints*, pp. arXiv–2603, 2026.

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 *2024 ieee/cvf conference on computer vision and pattern recognition (cvpr)*, pp. 8871–8879. IEEE, 2024.

Super Intelligence Team, Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, et al. FireRed-Image-Edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 2026.

Xiaoce Wang, Sifan Zhou, Kaifei Wang, Leli Xu, Xuerui Qiu, Tao He, and Ming Li. Why do DiT editors drift? plug-and-play low frequency alignment in vae latent space. *arXiv preprint arXiv:2605.08250*, 2026.

Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. OmniEdit: Building image editing generalist models through specialist supervision. In *International Conference on Learning Representations*, volume 2025, pp. 259–271, 2025.

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*, 2025a.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. OmniGen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Hang Xu, Xiaoxiao Ma, Guohui Zhang, Yu Hu, Siming Fu, Jie Huang, Lin Song, Haoyang Huang, Nan Duan, and Feng Zhao. AnchorEdit: Maintaining temporal consistency in multi-turn image editing via causal memory. *arXiv preprint arXiv:2606.11751*, 2026.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. ImgEdit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026a.

Yuxiao Ye, Haoran He, Fangyuan Kong, Xintao Wang, Pengfei Wan, Kun Gai, and Ling Pan. Edit-R2: Context-aware reinforcement learning for multi-turn image editing. *arXiv preprint arXiv:2606.05950*, 2026b.

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.

Liangbing Zhao, Zicheng Zhang, Xuecheng Nie, Luoqi Liu, and Si Liu. Cross-attention and seamless replacement of latent prompts for high-definition image-driven video editing. *Electronics*, 13 (1):7, 2023.

Liangbing Zhao, Le Zhuo, Sayak Paul, Hongsheng Li, and Mohamed Elhoseiny. From statics to dynamics: Physics-aware image editing with latent transition priors. *arXiv preprint arXiv:2602.21778*, 2026a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026b.

Wei Zhou, Xiongwei Zhu, Lingdong Kong, Bo Chen, Lei Zhang, Yongyuan Liang, Xiaoxia Hou, Ye Tian, Xian Sun, Yingshuo Wang, et al. On-policy self-distillation in diffusion models. *arXiv preprint arXiv:2608.24646*, 2026a.

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, et al. DanceOPD: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026b.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15792–15801. IEEE, 2025.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15329–15339, 2025.

A Implementation Details

Table 4 lists the full configuration for the three backbones. LoRA is applied to the attention and feed-forward projections of the diffusion transformer, and all other components remain frozen. Entries that need further definition are described below.

- **Query distribution.** The editing branch uses a training-time trajectory of N steps, independent of the sampling steps used for identity rollout. We sample two distinct query positions per update from a Beta distribution over the normalized trajectory index, following DanceOPD (Zhou et al., 2026b). Qwen-Image-Edit-2511 and FireRed-Image-Edit use $\mathrm{Beta}(5, 5)$, while FLUX.2-klein-base uses $\mathrm{Beta}(2, 5)$. Unlike DanceOPD, which reports $\mathrm{Beta}(5, 2)$ as the best setting, we find queries around the middle or the earlier, noisier part of the trajectory to work better.
- **Identity loss weight.** The identity loss is scaled by this factor relative to the editing loss, on top of the branch sampling ratio.
- **Drift threshold and patience.** Rollout drift is the mean absolute pixel difference between the current rollout state and the clean source image, computed in the 0 to 255 range. Only steps at the current rollout depth contribute. The depth increases by one turn once the drift stays at or below the threshold for *patience* consecutive such steps, after which the counter resets.

Table 4: **Implementation details of MT-OPSD across the three backbones.**

	Qwen-Image-Edit-2511	FireRed-Image-Edit	FLUX.2-klein-base
LoRA and optimization			
LoRA rank	32	32	32
LoRA α	64	64	64
Optimizer	AdamW	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}	1×10^{-4}
Learning-rate schedule	constant	constant	constant
Training samples	2,000	2,000	2,000
Training resolution	512^2	512^2	512^2
Identity rollout			
Sampling steps	30	30	50
Guidance scale	4.0	4.0	4.0
Editing branch			
Trajectory steps N	10	10	10
Query steps	2	2	2
Query distribution p_q	$\mathrm{Beta}(5, 5)$	$\mathrm{Beta}(5, 5)$	$\mathrm{Beta}(2, 5)$
Guidance scale	4.0	4.0	4.0
Objective and curriculum			
Identity loss weight	10	10	10
Initial rollout depth	2	2	2
Maximum rollout depth $K_{\max}$	10	10	10
Drift threshold	9.0	10.0	10.0
Patience	10	10	10

B Training Recipe

Algorithm 1 summarizes the training procedure of MT-OPSD. Each step draws a single source image and instruction, constructs the on-policy rollout state with the current student, applies one of the two branches of the training objective, and updates the rollout curriculum. Teacher promotion runs asynchronously and is described in Appendix C.

Algorithm 1 Training Procedure of MT-OPSD.

Require: Training set $\mathcal{D}$; pretrained editing model θ_0; identity instruction e_{id}
Require: Editing probability ρ; identity weight λ; query distribution p_q; initial and maximum rollout depth k_0, $K_{\max}$
Require: Drift threshold τ and patience P

Initialize student $\theta_S \leftarrow \theta_0$ and teacher $\theta_T \leftarrow \theta_0$
Initialize rollout depth $k \leftarrow k_0$ and counter $c \leftarrow 0$
while not converged **do**
  Sample $(I^{(0)}, e) \sim \mathcal{D}$
  Stage 1: Self-Generated Rollout
  $\tilde{I}^{(0)} \leftarrow I^{(0)}$
  for $j = 1, \ldots, k$ **do**
    $\tilde{I}^{(j)} \leftarrow G_{\theta_S}(\tilde{I}^{(j-1)}, e_{\mathrm{id}})$ ▷ no gradient
  end for
  Stage 2: Two-Branch Training
  Sample $b \sim \mathrm{Bernoulli}(\rho)$
  if $b = 1$ **then** ▷ editing branch
    $x_{t_N} \sim \mathcal{N}(0, \mathbf{I})$
    $\{x_{t_i}\}_{i=0}^{N} \leftarrow \mathrm{Rollout}(v_{\theta_S}; x_{t_N}, \tilde{I}^{(k)}, e)$ ▷ no gradient
    $\mathcal{Q} \leftarrow \mathrm{Sample}(p_q, 2)$
    $\mathcal{L}_{\mathrm{edit}} \leftarrow \dfrac{1}{|\mathcal{Q}|} \displaystyle\sum_{q \in \mathcal{Q}} \left\| v_{\theta_S}(\bar{x}_{t_q}, t_q, \tilde{I}^{(k)}, e) - v_{\theta_T}(\bar{x}_{t_q}, t_q, I^{(0)}, e) \right\|_2^2,$
    where $\bar{x}_{t_q} = \mathrm{sg}(x_{t_q})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{edit}}$
  else ▷ identity branch
    $\tilde{x}_0 \leftarrow \mathrm{Latent}(\tilde{I}^{(k)})$
    Sample t and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    $\tilde{x}_t \leftarrow (1-t)\tilde{x}_0 + t\epsilon$
    $\mathcal{L}_{\mathrm{id}} \leftarrow \left\| v_{\theta_S}(\tilde{x}_t, t, \tilde{I}^{(k)}, e_{\mathrm{id}}) - (\epsilon - \tilde{x}_0) \right\|_2^2$
    $\mathcal{L} \leftarrow \lambda \mathcal{L}_{\mathrm{id}}$
  end if
  Update θ_S by minimizing $\mathcal{L}$
  Stage 3: Adaptive Rollout Curriculum
  $d \leftarrow \mathrm{MeanAbsDiff}(\tilde{I}^{(k)}, I^{(0)})$
  if $d \leq \tau$ **then**
    $c \leftarrow c + 1$
  else
    $c \leftarrow 0$
  end if
  if $c \geq P$ and $k < K_{\max}$ **then**
    $k \leftarrow k + 1$; $c \leftarrow 0$
  end if
  Stage 4: Gated Teacher Promotion
  if the asynchronous evaluator promotes checkpoint $\theta^\star$ **then**
    $\theta_T \leftarrow \theta^\star$ ▷ Appendix C
  end if
end while

C Teacher Promotion Rule

Gate Set. Teacher promotion is determined using a held-out gate set of 23 ten-turn editing sessions. The sessions are constructed following the same procedure and source-image generator as LME-Bench, but use a disjoint set of source images. Fifteen sessions follow the standard LME-Bench composition of six local and four global edits, while the remaining eight contain local edits

only. The gate set is used solely for teacher selection and is excluded from all reported evaluation results. Checkpoints are evaluated using the same protocol and metrics described in Appendix D.2.

Asynchronous Evaluation. We save a student checkpoint every 50 training samples and evaluate each checkpoint asynchronously on four separate GPUs. Training proceeds without waiting for evaluation. Once a checkpoint is promoted, the decision is written to disk and picked up by the training process at its next polling step, where the current teacher is replaced by the promoted checkpoint.

Promotion Rule. Each candidate checkpoint is compared with the current teacher at turns 10, 8, and 5, in this order, giving priority to longer editing horizons. At turn k, we define $\Delta\mathrm{SR}_k$ as the number of gate sessions successfully completed through turn k by the candidate minus that of the teacher. Similarly, $\Delta\mathrm{CR}_k$ is defined as the number of sessions collapsed by turn k under the teacher minus that under the candidate. Thus, positive values favor the candidate for both quantities.

If $-1 \leq \Delta\mathrm{SR}_k < \alpha$ and $-1 \leq \Delta\mathrm{CR}_k < \beta$, the comparison is inconclusive at turn k, since neither quantity regresses by more than one session and neither reaches its promotion threshold, and the comparison proceeds to the next turn. Otherwise, the candidate is promoted if either

$$\begin{aligned} &\Delta\mathrm{SR}_k \geq \alpha \quad \text{and} \quad \Delta\mathrm{CR}_k \geq -1 && \text{(success-led)}, \\ &\Delta\mathrm{CR}_k \geq \beta \quad \text{and} \quad \Delta\mathrm{SR}_k \geq 1 && \text{(stability-led)} \end{aligned}$$

is satisfied, and the current teacher is retained if neither condition holds. If all three turns are inconclusive, no promotion is made. We set (α, β) to $(2, 5)$ for Qwen-Image-Edit-2511, $(2, 3)$ for FLUX.2-klein-base, and $(3, 5)$ for FireRed-Image-Edit. In practice, this rule typically results in one or two teacher promotions during each training run.

D LME-Bench and Evaluation Details

This section provides additional details on the construction of LME-Bench and its evaluation protocol.

D.1 Benchmark Construction

Source Images. We generate all source images with Z-Image-Turbo at 1024×1024 resolution using 9 sampling steps and no classifier-free guidance. Each image is seeded by its benchmark index to ensure exact reproducibility. The 100 sessions are evenly distributed across 10 categories: pets, wild animals, people, food, vehicles, interiors, nature, objects, plants, and urban scenes.

Session Structure. Each session contains 10 turns, with six local and four global edits. Local edits add, remove, replace, or modify a specific object or region, while global edits affect the image as a whole. We further divide global edits into *hard* globals, which change the rendering medium or apply a strong global transformation such as oil painting, anime, or grayscale, and *soft* globals, which adjust properties such as color temperature or brightness. Each session contains either three hard and one soft global edit, or two of each. The first turn is always a hard global edit, no two global edits are adjacent, and the final two turns are local. This arrangement introduces substantial changes early in the sequence while ensuring that later local edits operate on images that have already undergone several global transformations. Instructions never require access to an earlier image state or explicit comparison with previous turns; all referenced content must be identifiable from the immediately preceding image. Table 6 shows one complete session.

Validity Rules. Unlike single-turn editing, instruction validity in a multi-turn session depends on the preceding edits. For example, an object referenced by a later instruction may have been removed earlier, or the requested state may already hold. We therefore construct each session under an explicit set of rules that avoids non-informative turns, where either leaving the image unchanged can satisfy the instruction or success cannot be judged reliably. The rules are grouped into six families, summarized in Table 5. Among them, Already-True, Referent, and Interference specifically address dependencies across editing turns.

Table 5: **Validity rules used to construct LME-Bench.** The rules are grouped into six families according to the type of invalid or non-informative turn they prevent.

Family	Invalidity condition	Example rule
STRUCTURE	The chain violates the session schema	No two global edits are adjacent
VISIBILITY	The requested change is not visually verifiable	Counts in add/remove instructions are at most three
ALREADY-TRUE	The requested state already holds	A pose edit must request the opposite of the current state
REFERENT	The referenced object cannot be resolved	After a container is replaced, later turns may not refer to the old one
INTERFERENCE	Another turn would cancel or invalidate the edit	A grayscale or sepia edit must be the last global edit
FEASIBILITY	The requested change cannot be rendered reliably in the frame	An added object requires sufficient visible space at its implied size

Table 6: **Example ten-turn session from LME-Bench.**

Turn	Type	Instruction
1	hard global	Repaint the whole image as a textured oil painting with visible brushstrokes, keeping the dog and the room.
2	local	Change the wall behind the dog to a warm sage-green color.
3	hard global	Redraw the whole image in a bold anime illustration style with clean lines and cel shading.
4	local	Add a rubber ball on the floor beside the dog.
5	soft global	Shift the whole image to a cool blue-gray color temperature.
6	local	Make the dog lie down on the floor instead of sitting.
7	hard global	Convert the entire image to black and white, full grayscale with no color.
8	local	Change the wooden floor into pale marble tiles.
9	local	Add a small potted plant in the corner behind the dog.
10	local	Remove the rubber ball from the floor.

D.2 EVALUATION PROTOCOL AND METRICS

Execution. Each session is executed sequentially using the standard inference configuration of each backbone. At every turn, the model receives only the output from the preceding turn as its source image; the original image is never reintroduced.

Instruction Judging. We use GPT-4o to evaluate instruction execution at each turn. Given the source image, all instructions up to the current turn, and the corresponding intermediate outputs, the judge assigns integer scores from 0 to 10 for prompt following and consistency. Prompt following measures whether the requested change is realized relative to the immediately preceding image, rather than whether the current image merely contains the requested content. Consistency is defined according to the edit type. For local edits, it measures unintended changes outside the requested modification; for global edits, it measures whether the main subjects, their identities, and the composition remain consistent under the requested transformation. A turn is considered successful when both scores are at least 7.

Degradation Judging. A separate evaluation pass measures visual degradation independently of instruction execution. Given the source image, the current output, and the instruction history, the judge scores two aspects from 0 to 2: *content loss*, which measures whether the main subjects and scene remain recognizable, and *surface corruption*, which measures visual artifacts not requested by the instructions. Requested style changes are explicitly excluded from degradation. A turn is considered degraded when the two scores sum to at least 2. The complete judging prompts and scoring criteria are provided in Appendix D.3.

Metrics. SR@k denotes the fraction of sessions for which all turns up to turn k are successful. Following the early-stopping protocol of MSE-Bench, instruction evaluation stops after the first failed turn in each session. CR@k denotes the fraction of sessions that have collapsed by turn k. A session is considered collapsed once two consecutive turns are judged degraded and remains collapsed thereafter. Requiring two consecutive degraded turns reduces sensitivity to occasional judging noise. Both metrics are normalized by the total number of sessions.

D.3 JUDGE PROMPTS

Listing 1 shows the prompt used for instruction judging and Listing 2 the prompt used for degradation judging. Fields in braces are filled in per turn.

Listing 1: Prompt for instruction judging.

```
Assume you are an expert in evaluating multi-turn image editing. In
    this task, a user interacts with an image editing system across
    multiple turns. At the first turn, the user provides a source image
    and an editing prompt. The system returns the edited image. In each
    subsequent turn, the user supplies a new prompt, and the system
    generates a new image based on the output from the previous turn.
Your goal is to evaluate how successfully the editing instruction of
    the LAST turn (turn-{num_prompts}) has been executed.

You will be given {num_prompts} user editing prompts and {num_images}
    images: the first image is the original source image, and the next
    are the edited results from each turn for each prompt.
You should focus more on the last prompt and the last edited image, but
     you may also consider the previous prompts and images as context.

The {num_prompts} user editing prompts are: {editing_prompt}

Please follow these evaluation rules. For the LAST turn, assess TWO
    criteria by giving a reason, then assign an integer score from 0 to
    10 for each:

1) prompt_following: does the last edited image fulfill the last user's
     editing prompt? Judge the DELTA -- compare the last edited image
    against the IMMEDIATELY PRECEDING image (the second-to-last image
    provided) and verify that the SPECIFIC change the instruction asks
    for actually happened, instead of merely checking whether the final
    image happens to contain the described element. Apply this generally
     to the instruction type:
- Add a quantity ("add a second / another / one more X"): the COUNT of
    X must INCREASE by that amount versus the previous image (e.g. one X
     becomes two). An X that was already there does not count -- there
    must be a newly added one.
- Remove a quantity ("remove one / a X"): the count of X must DECREASE
    by one versus the previous image -- not "all of the X disappeared",
    and not "the X is unchanged".
- Change an attribute ("change X to <color/material/style>"): that
    specific X must actually take on the new attribute (the existing
    object changes), not merely that some object with that attribute now
     appears.
- Add or remove a target: the specific named object must actually
    appear (for add) or disappear (for remove) versus the previous image
    .
If the last image is essentially unchanged from the previous image
    where the instruction demanded a change, prompt_following must be
    low. (0 indicates the required change did not happen at all; 10
    indicates it was executed exactly as asked.)

2) consistency: are the parts that should be preserved still consistent
     with the input reference (the source image at the first turn, or
    the result image at the previous turn)?
First decide whether the last instruction is a LOCAL edit (adds,
    removes, or alters a specific object or region) or a GLOBAL edit (
    intentionally changes the whole image -- lighting, time of day,
    weather, season, color grade, or art style, e.g. "make it nighttime
    ", "convert to black and white", "make it a sunset scene").
- LOCAL edit: consistency rates the degree of overediting. 0 indicates
    the scene is needlessly different from the reference; 10 indicates a
     minimal yet effective edit that leaves everything else intact.
```

```
- GLOBAL edit: the whole image is SUPPOSED to change, so do NOT
    penalize the global change itself. Instead, consistency rates
    whether the main subjects, their identities, the composition, and
    the scene structure are preserved THROUGH the transform. 0 indicates
     the subjects or composition are lost, replaced, or distorted; 10
    indicates the same scene and subjects faithfully re-rendered under
    the requested global change.

3) Return your results in a JSON structure, following this format:
{"reason": "...", "prompt_following": int, "consistency": int}
```

Listing 2: Prompt for degradation judging.

```
You are auditing one frame of a multi-turn image-editing sequence for
    VISUAL DEGRADATION.

You are given the ORIGINAL SOURCE image first, and the CURRENT image
    last.
These editing instructions have been applied, in order, to get from the
     source to the current image:
{instructions}

Some of those instructions deliberately change how the picture LOOKS (
    repaint as pixel art, as watercolour, as an oil painting, convert to
     black and white, shift the colour temperature, change the season or
     weather). A picture that is a COMPETENT rendering in the requested
    style is NOT degraded, however far it is from a photograph. Blocky
    flat colour is correct for pixel art; visible brushstrokes are
    correct for an oil painting. Judge craftsmanship, not photographic
    realism.

Degradation is the opposite: damage no competent artist would produce.
    It includes heavy noise, rainbow or colour speckle, blur, melting or
     duplicated structure, extreme oversaturation, lost detail or
    sharpness, collapse into noise -- and also: UNREQUESTED flat colour
    patches replacing texture, hard-edged colour confetti or fragments
    scattered over the picture, a background that has stopped depicting
    a place and become an abstract pattern, and a subject reduced to a
    featureless silhouette.

Answer these two questions about the CURRENT image, then give a brief
    reason.

1. content_loss -- can the things that should be in the picture still
    be identified?
   0 = every main object and the setting are identifiable
   1 = at least one main object, or the setting, can no longer be made
    out
   2 = the picture no longer reads as the scene at all

2. surface_corruption -- is the picture's surface damaged in ways the
    instructions did not ask for?
   0 = none worth noting
   1 = clearly present, but the picture survives it
   2 = dominates the frame; the frame reads as noise or debris rather
    than a picture

Return JSON only: {"reason": "...", "content_loss": int, "
    surface_corruption": int}
```

E COMPARISON WITH VINCIE

Among the training-based multi-turn editing methods discussed in Section 2, VINCIE (Qu et al., 2026) is the only one for which we could obtain both public code and checkpoints. VINCIE learns

in-context editing from video and, in its native setting, conditions each turn on the full editing history, including the clean source image. We therefore evaluate VINCIE-7B both in this native full-history setting and under the previous-turn protocol used throughout our experiments, where each turn receives only the immediately preceding output. Table 7 compares these settings with the three backbones trained with MT-OPSD, all evaluated with previous-turn conditioning.

Table 7: **Comparison with VINCIE under different conditioning histories.** VINCIE is evaluated in its native full-history setting and under the previous-turn protocol used throughout our experiments. All MT-OPSD variants use previous-turn conditioning.

Method	LME-Bench					MSE-Bench		ImgEdit Overall
	SR@1	SR@5	SR@10	CR@5	CR@10	SR@1	SR@5	
VINCIE-7B (full history)	0.81	0.64	0.47	0.06	0.08	0.95	0.49	3.86
VINCIE-7B (previous turn)	0.84	0.15	0.09	0.17	0.46	0.91	0.30	3.86
Qwen-Image-Edit-2511 + MT-OPSD	**1.00**	**0.91**	0.44	0.01	**0.02**	0.98	0.51	4.49
FireRed-Image-Edit + MT-OPSD	1.00	0.88	**0.52**	**0.00**	0.03	**0.99**	**0.69**	**4.52**
FLUX.2-klein-base + MT-OPSD	0.99	0.76	0.38	0.00	0.04	0.95	0.49	4.28

Under the matched previous-turn protocol, VINCIE degrades sharply on LME-Bench: its SR@10 drops from 0.47 under full-history conditioning to 0.09, while CR@10 increases from 0.08 to 0.46. This suggests that access to the full editing history contributes substantially to its long-horizon performance. Under the same previous-turn protocol, MT-OPSD achieves higher SR@5 and SR@10 and lower collapse rates across all three backbones. Even compared with VINCIE in its native full-history setting, MT-OPSD obtains higher SR@5 and lower CR@5 and CR@10 on all three backbones, while the SR@10 comparison varies across backbones. On MSE-Bench, MT-OPSD matches or exceeds VINCIE at SR@5 on all three backbones. It also achieves higher single-turn ImgEdit scores than VINCIE.

F Additional Qualitative Results

Figures 5 and 6 show two additional LME-Bench sessions, each covering all three backbones and all comparison methods. Across both sessions, the base models and the three training-free baselines accumulate visible artifacts as editing proceeds, including high-frequency color speckle, fragmented texture, and loss of contrast, and both the dominant failure mode and the turn at which it appears differ across backbones. MT-OPSD follows the instruction at every turn on all three backbones, and its outputs remain largely free of such artifacts throughout the sequence.

G Qualitative Ablation Results

Figure 7 compares MT-OPSD with the variant trained without the identity branch on two LME-Bench sessions. Both models follow the requested edits throughout the sequence, consistent with the relatively high success rate of the variant in Table 3. The main difference lies in the preservation of content that is not targeted by the instructions. In the first session, the chef's appearance begins to drift from Turn-1 and changes progressively across subsequent edits, leading to a substantial identity shift by Turn-10. In the second session, the aircraft changes position and orientation from the first turn, with further layout drift accumulating over later turns. Without the identity branch, repeated editing introduces unnecessary changes to both subject identity and scene layout. In contrast, MT-OPSD follows the same editing sequence while better preserving subject identity and the overall composition across all ten turns. These observations are consistent with the lower single-turn performance of the variant on MSE-Bench and ImgEdit reported in Table 3.

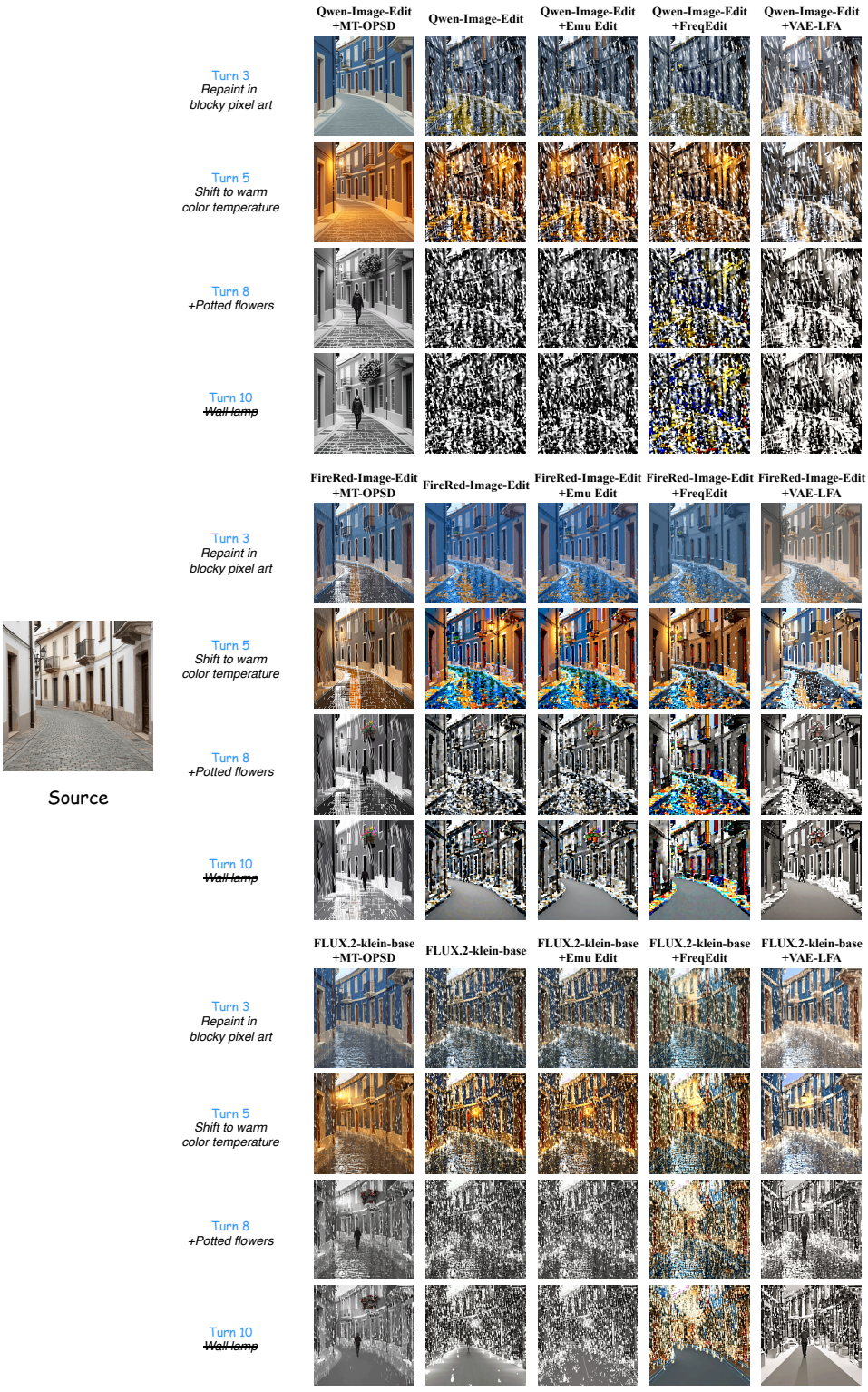

Figure 5: **Additional qualitative comparison on LME-Bench.** Selected turns of one editing session across three backbones.

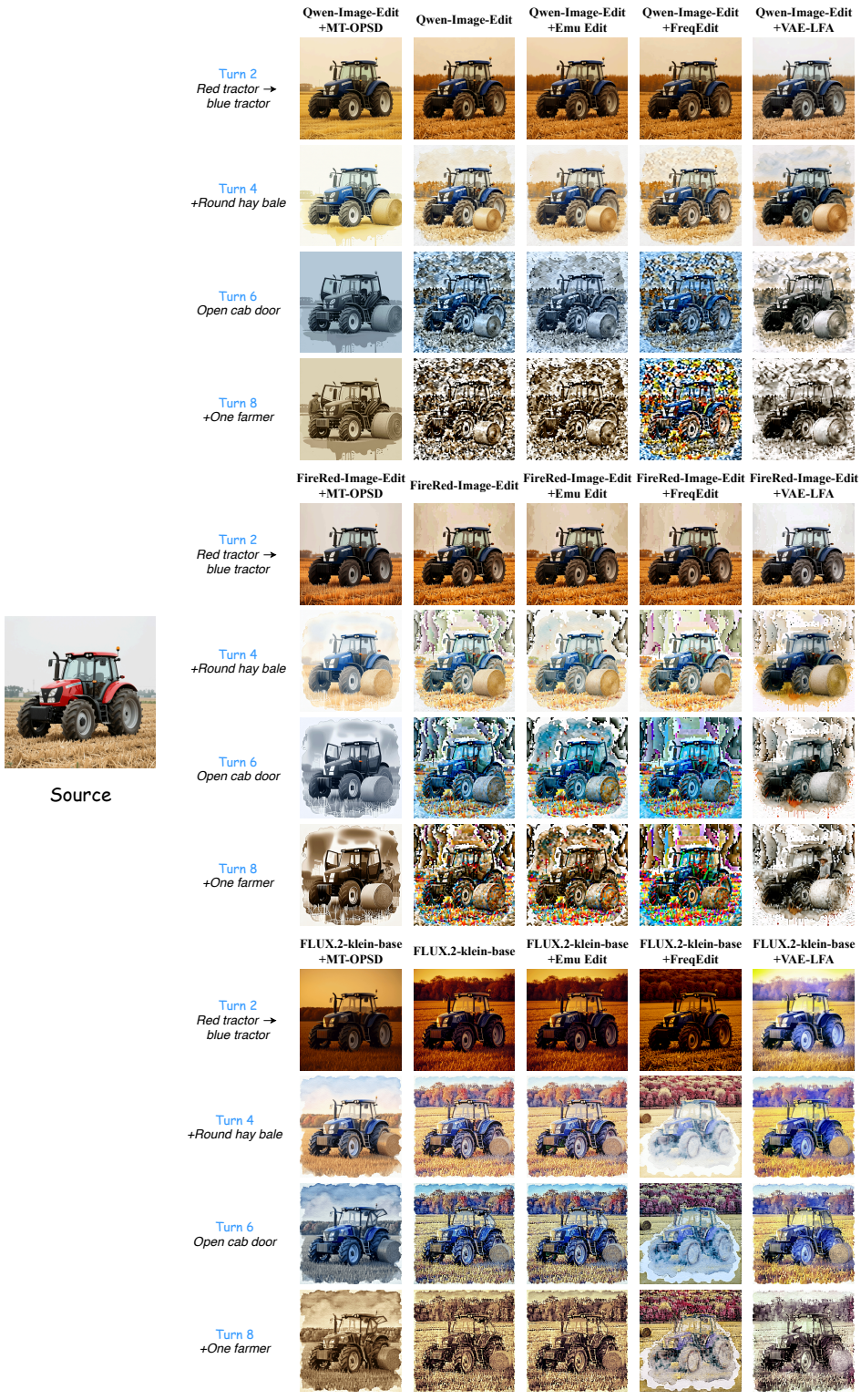

Figure 6: **Additional qualitative comparison on LME-Bench.** Selected turns of one editing session across three backbones.

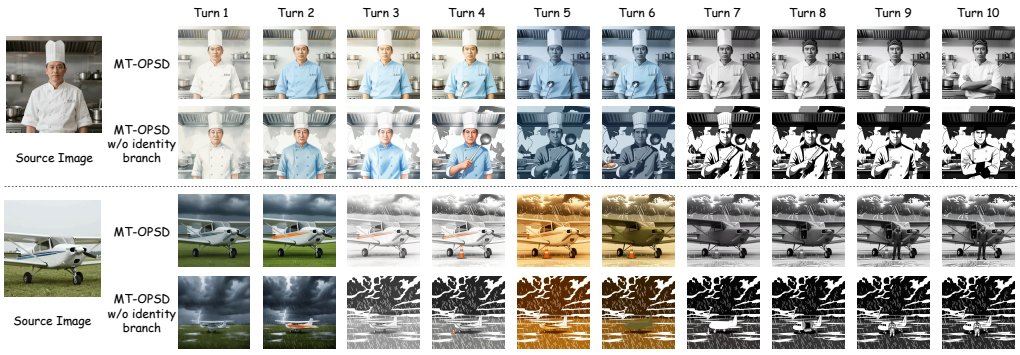

Figure 7: **Effect of the identity branch.** Removing the identity branch preserves instruction following but leads to progressive identity and layout drift across turns. MT-OPSD better preserves untargeted content while following the same ten-turn editing sequences.