ByteDance | Seed

DanceOPD: On-Policy Generative Field Distillation

Wei Zhou[1,2,‡], **Xiongwei Zhu**[1], **Zelin Xu**[1], **Bo Dong**[1], **Lixue Gong**[1], **Yongyuan Liang**[3], **Meng Chu**[4], **Leigang Qu**[2], **Lingdong Kong**[2], **Wei Liu**[1,†], **Tat-Seng Chua**[2]

[1]ByteDance Seed, [2]NUS, [3]UMD, [4]HKUST

[‡]Work done at ByteDance Seed, [†]Corresponding author

Abstract

Modern image generation demands a single model that unifies diverse capabilities, including text-to-image (T2I), local editing, and global editing. However, these capabilities are rarely naturally aligned and often conflict. For instance, editing tends to degrade T2I performance, while global and local editing interfere with each other. Consequently, effectively composing these capabilities has become a central challenge for image generation model training. To tackle this, we introduce DanceOPD, an on-policy generative field distillation framework for flow-matching models that routes each sample to one capability field, queries one low-noise student-induced state, and trains with a simple velocity MSE objective. With each capability source defined as a velocity field over the shared flow state space, the student learns from fields queried on its own rollout states to compose expert capabilities. This formulation also absorbs operator-defined fields such as classifier-free guidance. Comprehensive experiments on T2I, editing, realism-field absorption, and CFG absorption show that our approach improves multi-capability composition, strengthening target capabilities while preserving anchor generation quality. We believe this work establishes a practical route for generative field distillation in flow-matching models.

Project Page: DanceOPD.github.io **GitHub Repo:** github.com/worldbench/DanceOPD
Correspondence: Wei Liu (E-mail: liuwei.jikun@bytedance.com)
Date: June 26, 2026

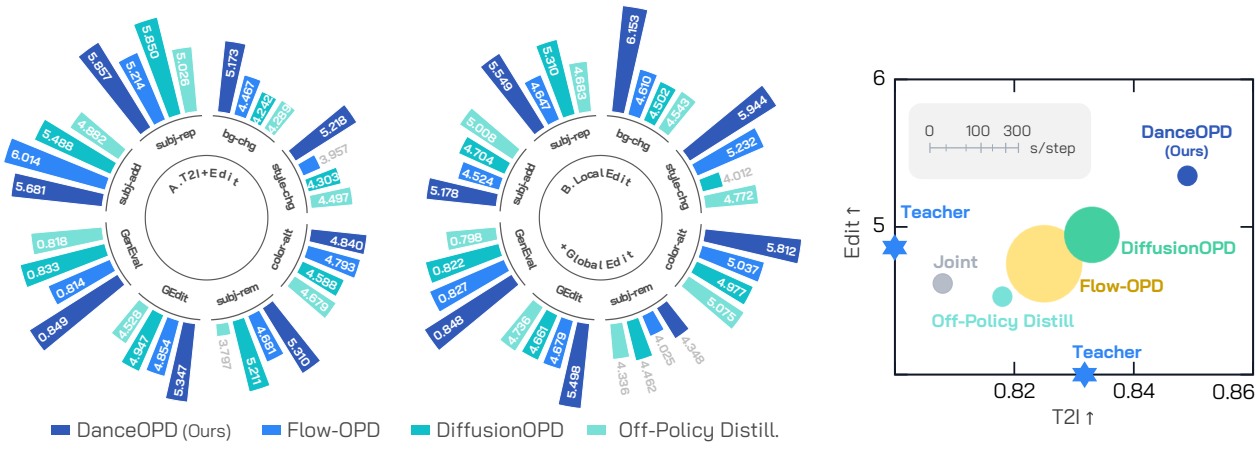

Figure 1 DanceOPD Improves Multi-Capability Composition. Left: Per-metric performance for the two composition settings, compared with representative baselines. **Right:** Editing×T2I capability space for their composition, with marker size denoting per-step training cost. DanceOPD achieves a better composition while incurring a lower cost.

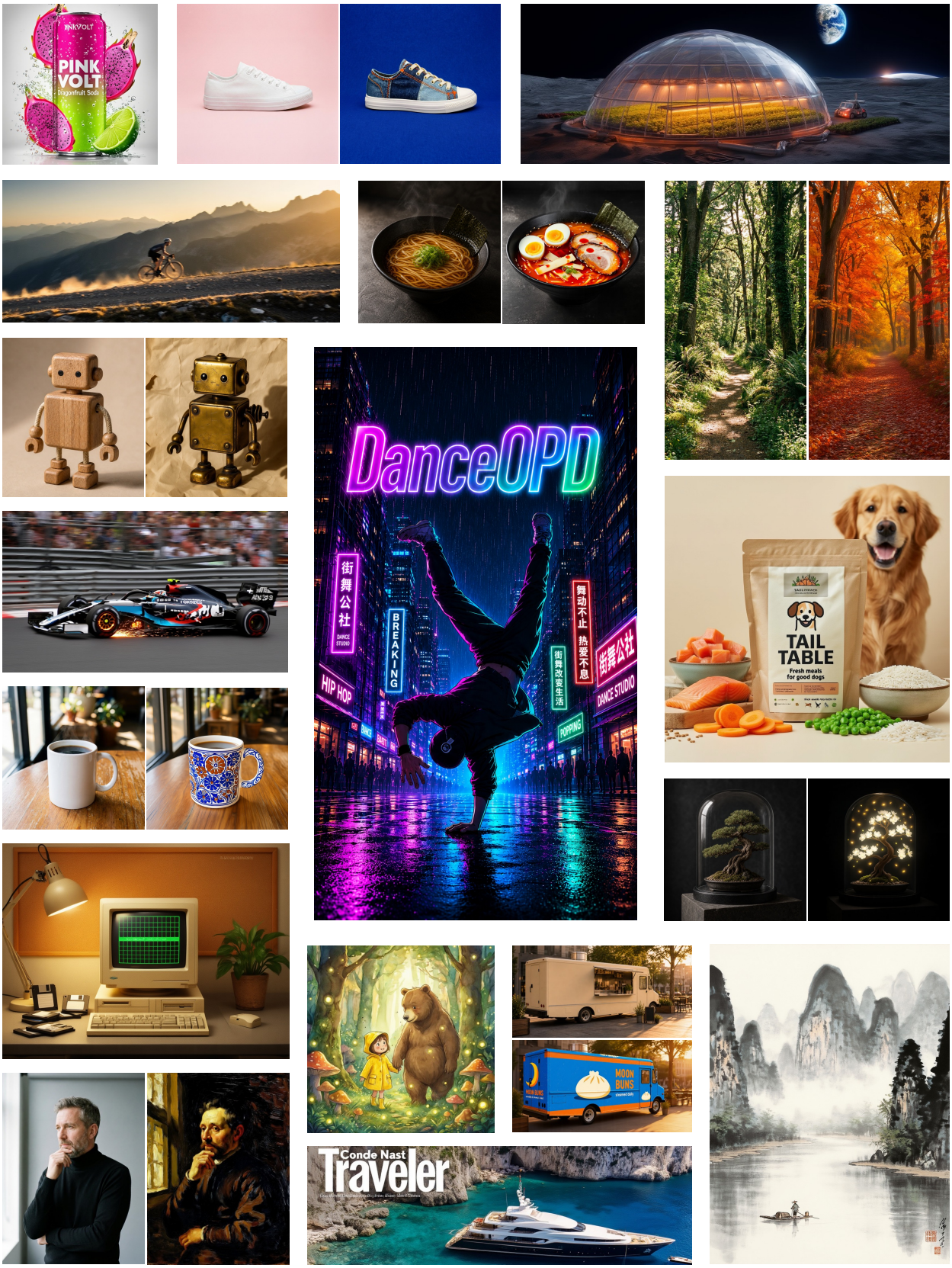

Figure 2 Qualitative Examples from DanceOPD. After training with DanceOPD, the resulting student supports diverse text-to-image and editing behaviors while retaining strong original generative capabilities.

Contents

1 Introduction

Modern image generation has rapidly advanced through flow matching models [9, 21, 37, 72, 74, 77]. A single deployed model is increasingly expected to jointly support diverse capabilities, including text-to-image generation (T2I), local editing, and global editing [8, 16, 106]. These capabilities, however, are not naturally compatible. T2I rewards open-ended visual quality and prompt following; local editing requires preserving the input while applying precise changes; and global editing intentionally changes broad appearance statistics such as style, color, or layout. Naively optimizing them together often leads to capability interference: editing can erode T2I performance, while local and global editing can pull the model toward conflicting preservation and transformation behaviors. Consequently, how to effectively compose these capabilities has become a central challenge for the image generation model community.

Existing capability-combination paradigms only partially address this goal. Data mixing or joint training tends to dilute capability-specific supervision and can suffer from multi-task gradient conflict [13, 42, 44, 68, 85, 98]; parameter-space merging and adapter composition typically yield compromise solutions [3, 28, 73, 80, 101]; and inference-time score composition leaves the composition external to the deployed student [6, 20, 59]. While useful, these approaches do not directly resolve the state-dependent question:

How should a model effectively compose multiple generative capabilities?

Building on recent formulations that interpret task abilities as velocity fields in flow-matching image generators [4, 18, 55, 61, 94], we adopt a field-based perspective in which each frozen capability source defines a velocity field over a shared generative state space. Under this view, capability composition reduces to a field-query problem with three coupled choices: which capability field should supervise a given sample, where in the state space the field should be queried, and how many states from the student rollout should be used for supervision. This formulation aligns naturally with on-policy distillation (OPD) [1, 30, 45, 79], in which the teacher supervises states produced by the current student rather than fixed offline trajectories.

These three choices directly determine the main alignment challenges in multi-capability distillation. For the choice of target field, linearly combining several frozen fields within a single sample target can produce a supervision direction that does not correspond to any well-defined capability query, causing target-field ambiguity and losing the semantic identity of each capability. For the choice of query state, evaluating the field on data states, teacher trajectories, or other off-policy states leaves the student under-supervised on the states it actually visits during generation, causing state-distribution mismatch and a gap between training and inference. For the choice of trajectory supervision, drawing dense targets from multiple states along the same rollout overcounts highly correlated supervision signals, since these states share the same prompt, noise seed, student dynamics, and path history. This causes trajectory-query correlation and can bias the resulting gradient.

Guided by these diagnoses, we introduce **DanceOPD**, an on-policy generative field distillation framework whose three design choices correspond to the three challenges above.

- To resolve target-field ambiguity, our method performs hard-routed sample-wise field matching: each sample is dispatched to exactly one frozen capability field, preserving its semantic identity.
- To resolve state-distribution mismatch, the routed field is queried on a stop-gradient state drawn from the current student rollout, aligning supervision with the student's own visitation distribution.
- To resolve trajectory-query correlation, we use a single semantic-side low-noise query per sample, located in the regime where capability-specific information is most concentrated, thereby avoiding correlated within-trajectory samples.

The student is updated by a plain velocity MSE loss, which is the natural local regression objective for deterministic velocity fields: under a local Gaussian transition view, KL-style field matching reduces to a weighted MSE form, as derived in Appendix Sec. 7. The same formulation subsumes operator-defined fields; classifier-free guidance, in particular, can be cast as a guided velocity field and absorbed into the student under the identical objective [36].

We evaluate our approach across two capability-composition settings and two field-absorption cases: T2I and editing composition, local and global editing composition, realism-field absorption with base T2I preservation, and CFG absorption. The main results show that hard-routed on-policy field matching improves multi-capability composition, strengthening target edit or realism-oriented fields while preserving base generation quality. In T2I and editing composition, DanceOPD improves GEditBench over the best reproduced OPD baseline by 8.1% and over the edit source by 8.5%, while slightly exceeding the T2I source on GenEval. In local and global edit composition, our model improves over the best competing composition baseline by 16.1% and over the local edit source by 7.9%, with GenEval above all compared composition baselines.

For realism-field absorption, our model improves the realism reward over off-policy distillation by 9.9% and closes 85.3% of the student-to-teacher reward gap while maintaining the T2I score within 0.1% of off-policy distillation and above the student anchor by 7.6%. For CFG absorption, the best measured composition improves over train-only absorption by 7.6% and over eval-only CFG by 1.4%, while over-guided composition drops substantially. The diagnostic studies further support the method design: hard routing improves over soft all-teacher mixing by 15.2% under MSE and 10.6% under KL; semantic-side low-noise queries improve over median- and high-noise queries by 23.7% and 19.5%; dense same-step accumulation drops by 22.8%, and SDE-style decorrelation partially rescues the stress case with an 18.4% gain but remains 8.6% below the single-query default; and plain velocity MSE improves over the main weighted alternatives by 2.8% to 4.5%.

Our contributions can be summarized as follows:

- We formulate multi-capability image generation as on-policy generative field distillation and introduce DanceOPD, a hard-routed, semantic-side velocity-matching method on student-visited states.
- We identify three query-induced alignment challenges, including target-field ambiguity, state-distribution mismatch, and trajectory-query correlation, and show how they motivate our design.
- Experiments across T2I and editing composition, realism-field absorption, and CFG absorption, together with comprehensive ablations, confirm the effectiveness of DanceOPD.

2 Related Work

2.1 Multi-Capability Composition

Modern image generation increasingly requires a single deployed model to support multiple capabilities, including open-ended text-to-image generation [9, 72, 77], local editing that preserves the source image while applying targeted changes [5, 8, 106], global editing that changes broad appearance, layout, or style statistics [16, 17], and style-specialized or personalized generation [28, 80, 86]. These capabilities are not naturally compatible: text-to-image generation emphasizes open-ended visual quality and prompt following, local editing emphasizes source preservation under precise changes, and global or style-oriented editing emphasizes broader transformation. Therefore, multi-capability composition is not merely a matter of covering more generation and editing methods; it asks how one deployed student can strengthen a target capability while preserving the anchor capability. Existing multi-capability composition methods often incur degradation in individual capabilities when combining these abilities. Data mixing or joint training can dilute capability-specific supervision and expose gradient conflict [13, 85]; parameter-space merging often yields compromise solutions [101, 102], as do personalization and adapter composition [52, 86] and adapter-fusion analyses [12, 67]; and inference-time score composition leaves the composition outside the deployed student [6, 59]. In contrast, DanceOPD treats each frozen capability source as a velocity field [55, 61] and learns a single student through hard-routed on-policy field matching, thereby combining the strengths of different teacher models and even achieving performance beyond individual teachers, as shown in Fig. 1. Table 1 summarizes how this positioning differs from prior on-policy distillation methods.

2.2 On-Policy Distillation

On-policy distillation addresses the mismatch between fixed training states and states induced by the current student. It differs from standard knowledge distillation and diffusion distillation, which mainly target model compression, sampler compression, trajectory training, consistency training, or distribution

Table 1 Comparison with OPD Methods. To our knowledge, DanceOPD is the only method that combines flow-matching OPD, multi-capability composition, design-space analysis, and functional field absorption.

Method	Domain	Teacher Signal	Objective	FM-OPD	Multi-Cap.	Design Study	Func. Absorp.
MiniLLM [30]	LLM	logits	reverse KL	–	–	–	–
GKD [1]	LLM	logits	forward KL	–	–	–	–
AOPD [45]	LLM	logits and top-K	asymmetric KL	–	–	–	–
G-OPD [103]	LLM	scalar feedback	policy optimization	–	○	–	–
StableOPD [64]	LLM	scalar reward	PPO with KL anchor	–	○	–	–
ROPD [23]	LLM	rubric reward	reward optimization	–	○	–	–
DiffusionOPD [53]	Flow	velocity field	KL or MSE-style	✓	○	○	–
D-OPSD [46]	Diffusion	predicted distribution	self-distillation	○	–	–	–
Flow-OPD [24]	Flow	dense scalar reward	PPO clip-min	✓	task-routed	–	–
★ DanceOPD	Flow	routed velocity field	MSE	✓	✓	✓	✓

matching [35, 48, 65, 81, 82, 93, 104]. In language models, KL-based OPD methods match teacher token distributions on student-generated sequences [1, 30, 45], while reward-based variants optimize scalar feedback with policy-gradient or PPO-style objectives [23, 64, 103]. Recent work further shows that rollout position affects OPD stability and efficiency [50, 111], suggesting that the location of teacher supervision along a student rollout is itself an important design choice.

Recent flow-model OPD methods instantiate related ideas for generative post-training. DiffusionOPD performs on-policy velocity matching, D-OPSD studies on-policy self-distillation, and Flow-OPD uses dense reward optimization with PPO-style clipping [24, 46, 53]. Our focus is complementary: we study how T2I, editing, realism, and guidance-related generative capabilities can be composed through on-policy field distillation in flow-matching models. This requires deciding which teacher field supervises each sample, where the field should be queried, and how many rollout states should contribute supervision. We therefore compare KL-style objectives, teacher routing, query states, and dense versus single-query supervision, and find that one low-noise semantic query on the student's rollout is an effective and substantially cheaper default. We also evaluate realism-field absorption, CFG absorption, and mismatched training rollout and evaluation sampler step counts, which reflect practical post-training pipelines.

2.3 Generative Field Distillation

Flow-matching formulations view generation as learning a velocity field over a continuous state space, building on diffusion and score-based generative modeling foundations [4, 18, 37, 55, 61, 94]. This field view is useful for capability composition because a frozen T2I model, edit model, realism-oriented model, or guidance operator can all be treated as sources of local velocity supervision on a shared state space. Prior work has explored related ingredients: score or field composition combines multiple generative signals at inference time [6, 20], multi-task optimization studies how to handle conflicting gradients during joint training [13, 44, 56, 85, 107], and on-policy imitation or distillation emphasizes supervision on states visited by the current student [24, 53, 79]. These directions motivate our formulation but do not specify how multiple capability fields should be queried during distillation. DanceOPD addresses this missing design problem with hard field routing, student-induced query states, and one semantic-side low-noise query.

3 Approach

We formulate multi-capability image generation as on-policy generative field distillation. Given several frozen capability sources, our goal is to train one student to query and imitate these sources as velocity fields on a shared generative state space [4, 55, 61], rather than combining them through static parameter interpolation or data-ratio tuning [42, 98, 101]. This formulation turns capability composition into a field-query problem with three coupled choices: which capability field should supervise a sample, where in the state space the field should be queried, and how many states from the student rollout should be used for supervision. Fig. 3 summarizes the resulting training query. Sec. 3.2 to Sec. 3.4 instantiate these choices through hard-routed field

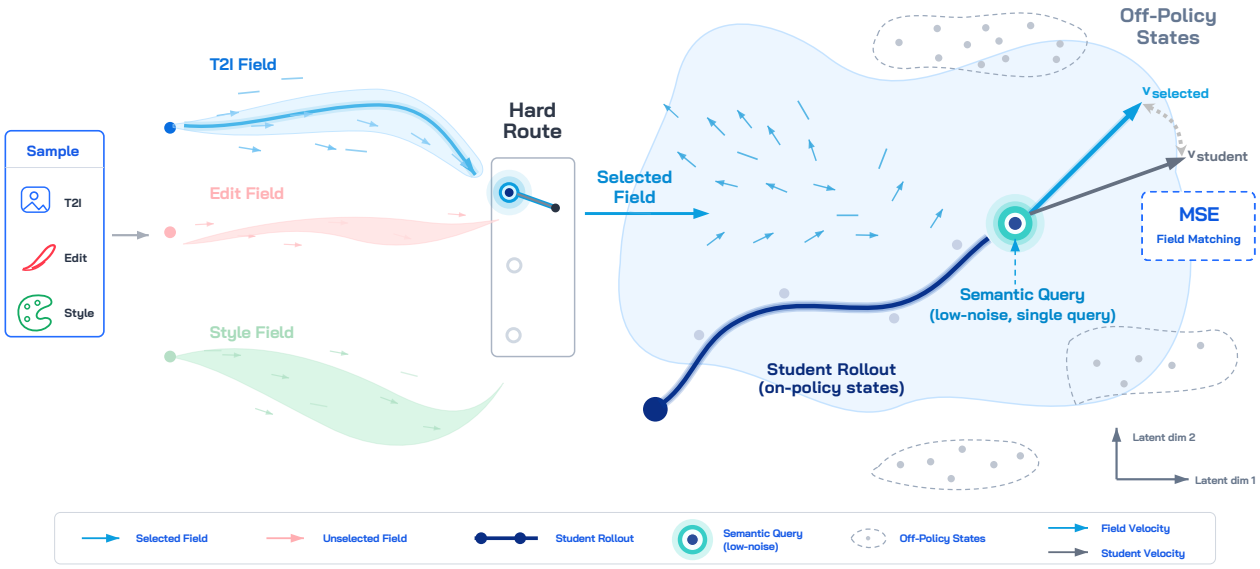

Figure 3 Conceptual Illustration of DanceOPD. For each sample, DanceOPD hard-routes supervision to one capability field, queries that field at a single low-noise state on the current student rollout rather than off-policy states, and aligns the student velocity to the selected field velocity with MSE.

selection, on-policy student-state querying, and semantic-side single-query supervision, addressing target-field ambiguity, state-distribution mismatch, and trajectory-query correlation, respectively; Sec. 3.5 then gives the local matching objective.

3.1 Preliminary

Assume we are given $M \geq 2$ frozen capability sources that are defined on the same generative state space and expose compatible velocity predictions. The sources may correspond to different tasks or behaviors, such as text-to-image, image editing, local attribute editing, global editing, or style-specialized generation. We denote the trainable student by v_θ, whose parameters are updated during distillation, and the frozen capability sources by $\{v_m\}_{m=1}^{M}$. For each source, $\mathcal{D}_m$ denotes the corresponding route-specific training distribution, x denotes the route-specific data payload, and p_T denotes the initial-noise distribution. This source-selection view is related to conditional expert routing and multi-task gating [43, 66, 87].

A naive unified model often suffers from capability dilution: joint training averages supervision signals, weight merging assumes approximate parameter-space linearity, and data-ratio tuning merely moves the model along a trade-off curve [13, 56, 85, 107]. Our goal is different. We define the desired composition outcome operationally: under the reported metrics, a single model should improve the target capability while preserving the anchor capability. This does not claim global Pareto optimality over all possible training mixtures, but specifies the target region for capability composition.

We view each frozen source as defining a capability-specific velocity field:

$$v_m(z_t, t, c), \qquad m \in \{1, \ldots, M\}, \tag{1}$$

where z_t is a flow state at time t and c denotes the conditioning information, such as a text prompt, source image, edit instruction, or style condition. This abstraction is independent of how the source is obtained: a source can be a pretrained model, a fine-tuned model, or a task-specialized branch.

Under this view, capability composition becomes a field-alignment problem: which capability field should the student imitate, and at which states should the field be queried? This yields three design questions that structure the rest of the method: which field supervises each sample, where the field is queried, and how many trajectory states are used for supervision.

3.2 Hard-Routed Sample-Wise Field Matching

The first design question is which field should supervise each sample. A naive alternative is to average several fields inside the same sample target, as in soft multi-teacher or knowledge-amalgamation variants [35, 88]. However, such an averaged target may no longer correspond to any well-defined capability query, especially when the fields encode different tasks or visual behaviors. To keep the target semantically meaningful, we route each training sample to exactly one capability field. A sample drawn from the text-to-image partition queries the T2I field; an edit sample queries the edit field; a local-edit or style-edit sample queries the corresponding field. Let $\pi(m)$ denote the route probability over active capability buckets. Formally, this is:

$$m \sim \pi(m), \qquad (x, c) \sim \mathcal{D}_m, \tag{2}$$

We do not tune π; unless otherwise stated, it is uniform over the active capability buckets, so two-bucket composition uses a 1:1 route ratio and the three-bucket diagnostics use a 1:1:1 route ratio. We then define the routed target field as:

$$u_m(z, t, c) = v_m(z, t, c). \tag{3}$$

This routing preserves the semantic identity of each capability query, in contrast to multi-teacher distillation schemes that combine teacher signals before or within a student update [27, 105]. It avoids injecting unrelated capability directions into a single sample target, and it avoids forcing several potentially conflicting fields to be averaged inside one sample-level supervision signal. Capability composition is therefore achieved statistically across routed updates, while each individual query remains semantically well-defined.

This design is also consistent with observations in multi-task optimization: when task gradients are poorly aligned, averaging them in the same optimizer step can produce a compromise direction rather than progress for every task [13, 44, 56, 85, 107]. Our method, therefore, uses hard-routed sample-wise field matching as the default capability-composition estimator and studies same-step multi-field accumulation only as an ablation.

3.3 On-Policy Field Querying

The second design question is where the routed field should be queried. A student that composes several capabilities generally follows a trajectory different from any individual frozen source. Therefore, aligning fields only on data states or on teacher-induced trajectories can leave a mismatch between training and testing: at inference, the student visits states z_t^θ that are not necessarily covered by the off-policy query distribution.

DanceOPD instead queries the capability fields on the student's own rollout. Let:

$$z_{0:T}^\theta = \mathrm{Rollout}(v_\theta; z_T, c), \qquad z_T \sim p_T, \tag{4}$$

be the trajectory induced by the current student from initial noise z_T. We sample a semantic coordinate s and its physical timestep $t = t(s)$, set the query state to $\bar{z}_t = \mathrm{sg}(z_t^\theta)$, query the frozen capability field as $v_m(\bar{z}_t, t, c)$, and train the student to match that field locally. This is the flow-model counterpart of on-policy distillation: the frozen source supervises the student on the distribution induced by the student itself, addressing the same covariate-shift issue that motivates on-policy imitation learning and recent OPD methods [1, 24, 53, 79]. The stop-gradient query means that the update differentiates the local velocity prediction, not the entire rollout solver. Appendix Sec. 7.2 gives the corresponding estimator and mismatch bound.

3.4 Semantic-Side Single Query

The third design question is how many states should be queried from the student trajectory. A natural extension is to query many states from the same rollout. However, dense trajectory queries can fail because multiple states from one rollout are correlated: they share the same initial noise, prompt, conditioning, student dynamics, and path history. Thus, their gradients are not independent, and adding more states does not necessarily provide more independent supervision, echoing multi-task optimization settings where gradient aggregation itself can become the bottleneck [14, 58, 70].

Algorithm 1 One DanceOPD Training Step. A sample is routed to one capability field, queried on a semantic-side student state, and updated by velocity matching.

Require: Frozen capability fields $\{v_m\}_{m=1}^{M}$, student v_θ, route probabilities π, query distribution q_{sem} over semantic coordinate s.
Ensure: Updated student parameters θ.

A. Route One Capability Query

$m \sim \pi(m), \quad (x, c) \sim \mathcal{D}_m$ ▷ preserve sample identity

B. Query On The Student Trajectory

$z_T \sim p_T, \quad z^{\theta}_{0:T} \leftarrow \text{Rollout}(v_\theta; z_T, c)$

$s \sim q_{\text{sem}}(s), \quad t \leftarrow t(s), \quad \bar{z}_t \leftarrow \text{sg}(z^{\theta}_t)$ ▷ one semantic-side state

$u \leftarrow v_m(\bar{z}_t, t, c)$ ▷ frozen routed field

C. Match The Local Velocity Field

$\mathcal{L} \leftarrow \left\| v_\theta(\bar{z}_t, t, c) - u \right\|_2^2$

$\theta \leftarrow \text{OptStep}(\theta, \nabla_\theta \mathcal{L})$

These considerations motivate a semantic-side single query. Let $s \in [0, 1]$ be a normalized semantic-side rollout coordinate, and let q_{sem} be a query distribution biased toward the low-noise side of the trajectory. Rather than supervising the whole trajectory, DanceOPD samples one high-information state:

$$K = 1, \qquad s \sim q_{\text{sem}}(s), \quad t = t(s), \tag{5}$$

High-noise states can contain coarse structure, but they are often dominated by generic denoising and have a lower density of capability-specific signal. Low-noise states are closer to the final image and concentrate style, aesthetics, local attributes, and task-specific edit information. This view is consistent with prior diffusion editing and distillation studies showing that timestep and noise-schedule choices strongly affect the fidelity and editability trade-off and that semantic-related timesteps are especially important for image manipulation [11, 33, 54, 69, 96]. Thus, a single low-t query provides a high signal-to-noise capability supervision while avoiding correlated trajectory samples.

Dense-query variants are useful as diagnostics. If dense-query degradation is partly caused by trajectory correlation, then decorrelating the rollout should partially mitigate this failure in dense-query stress cases. We test this prediction with stochastic rollout noise, which reduces query correlation. The detailed SDE formulation and the correlation-decay argument are deferred to the appendix.

3.5 Objective Design

Once the routed field, student-induced query state, and semantic-side query time are fixed, the remaining choice is the local matching objective. Our default objective is plain velocity MSE on the routed, on-policy query. With the stop-gradient query state $\bar{z}_t = \text{sg}(z^{\theta}_t)$, we optimize:

$$\mathcal{L}_{\text{DanceOPD}} = \mathbb{E}_{m\sim\pi,\,(x,c)\sim\mathcal{D}_m,\,z_T\sim p_T,\,s\sim q_{\text{sem}}} \left[\left\| v_\theta(\bar{z}_t, t, c) - v_m(\bar{z}_t, t, c) \right\|_2^2 \right], \qquad t = t(s). \tag{6}$$

The target is a deterministic velocity field, making MSE the natural regression objective under an isotropic Gaussian velocity-likelihood view [37, 55]. KL-style velocity matching reduces to a weighted MSE form under the same view; we provide the derivation in Appendix Sec. 7.1. This statement is limited to the local velocity-matching subproblem and should not be interpreted as an information-theoretic ceiling on downstream task metrics. Empirically, Sec. 4.3 shows that plain MSE is the most stable and best-performing objective among the tested alternatives, including KL weighting, timestep-weighted objectives, and score-style and consistency-style surrogates.

The same formulation also applies to operator-defined fields. For example, let $v_\emptyset$ and v_{cond} denote the unconditional and conditional velocity predictions. Classifier-free guidance with guidance scale α defines a guided velocity field, that is:

$$v_\alpha(z_t, t, c) = v_\emptyset(z_t, t) + \alpha\big(v_{\text{cond}}(z_t, t, c) - v_\emptyset(z_t, t)\big), \tag{7}$$

which can be treated as an additional capability field and absorbed into the student by the same MSE objective [36].

Fig. 5 further illustrates how the same field-matching objective handles material, environment, and lighting edits. In the apple example, DanceOPD transforms the red apple into a clear crystal-like object while keeping the leaf, tabletop, and studio lighting layout. In the piano example, DanceOPD introduces an underwater environment with caustic light and bubbles while retaining the grand-piano structure. In the leather-armchair example, the chair geometry and leather appearance remain visible under a stronger cinematic spotlight, whereas several baselines either over-expose the object or make the scene too dark to preserve the chair details. These cases support the role of routed, semantic-side supervision: the edit field can change the requested visual factor without forcing unrelated changes to the source object or scene layout.

4 Experiments

We evaluate DanceOPD primarily on Z-Image [9] for composing frozen capability fields into one student, and use SD3.5-M for the realism-field absorption setting, with implementation details provided in the appendix. The experiments are organized around three questions: whether a single student can compose heterogeneous capability fields without collapsing into a compromise between them; which field-routing and query choices are needed when multiple capability fields supervise the same student; and how the main training choices, including student-induced query states, semantic-side single querying, rollout discretization, initialization, and plain velocity MSE, affect performance. Sec. 4.1 reports the main results for capability composition and field absorption under a unified protocol, Sec. 4.2 studies the multi-teacher extension and routing diagnostics, and Sec. 4.3 ablates query and objective design. Across all settings, the main finding is that DanceOPD improves the target capability while preserving the anchor capability, whereas joint training, weight merging, soft teacher mixing, and dense same-step supervision tend to reintroduce capability interference.

4.1 Main Results

To evaluate the effectiveness of DanceOPD, we consider four settings that cover both capability composition and field absorption: T2I and editing composition, local and global editing composition, realism-field absorption with base T2I preservation, and CFG absorption. For the two editing-related composition settings, we use GenEval to measure general text-to-image generation ability and GEditBench to measure general editing ability [29, 60]. For realism-field and CFG absorption, we use diagnostic metrics matched to the absorbed target fields while also monitoring preservation of the anchor generation capability. Across these settings, DanceOPD consistently strengthens the target capability while preserving the anchor capability, showing that routed on-policy field matching can compose and absorb generative fields without collapsing into a simple compromise between sources.

A. T2I and Edit Composition. We first test whether DanceOPD can add general editing ability to a T2I student while preserving its base generation capability. DanceOPD improves the GEditBench average over the best reproduced OPD baseline by 8.1% and over the edit source by 8.5%, while improving GenEval overall over the T2I source by 2.0% and over the strongest composition baseline by 1.6%. The gains are especially clear on edit categories that require larger visual changes. Compared with DiffusionOPD, DanceOPD improves background change by 21.9%, style change by 21.3%, and color alteration by 5.5%. These results indicate that routed on-policy field matching can integrate the edit field into the student without eroding the base generation field. The qualitative examples show the same pattern, where DanceOPD follows scene, style, material, and object-level edit instructions while retaining prompt-following T2I generation. Fig. 2, Fig. 12, Fig. 13, Fig. 5, Fig. 6, and Fig. 14 show the same pattern for T2I and Edit Composition: DanceOPD follows scene, style, material, and object-level edit instructions while retaining prompt-following T2I generation.

B. Local and Global Edit Composition. We then evaluate whether one student can compose local editing, which emphasizes preservation, with global editing, which requires broader visual transformation. DanceOPD improves the GEditBench average over the best competing composition baseline by 16.1% and over the local edit source by 7.9%, while improving GenEval overall over the strongest composition baseline by 2.5%. The improvement is concentrated in categories that require global or attribute-level changes. Compared with the

Table 2 Main Multi-Capability Composition Results with Two Different Settings. We compare DanceOPD with Joint Training, weight merging, off-policy distillation, and other OPD baselines on GEditBench-EN [60] for image editing and GenEval [29] for text-to-image. DanceOPD achieves better capability composition.

Method	GEditBench-EN [60]							GenEval [29]						
	Subj-Add	Subj-Rep	Bg-Chg	Style-Chg	Color-Alt	Subj-Rem	Avg	Sing	Two	Cnt	Col	Pos	C-Attr	Overall
Base Model														
T2I	–	–	–	–	–	–	–	0.950	0.939	0.938	0.947	0.520	0.700	0.832
Edit	6.033	5.417	4.490	3.923	4.889	4.828	4.930	0.838	0.828	0.713	0.840	0.580	0.470	0.711
Local Edit	5.555	5.742	4.856	3.817	4.581	6.017	5.095	0.988	0.929	0.813	0.862	0.600	0.570	0.793
Global Edit	3.119	4.414	4.040	5.209	4.287	1.433	3.750	0.950	0.939	0.838	0.872	0.600	0.650	0.808
A. T2I and Edit Composition														
Joint Training	5.386	5.627	4.283	3.688	3.624	5.093	4.617	0.975	0.929	0.963	0.894	0.540	0.550	0.808
Weight Merge	–	–	–	–	–	–	–	1.000	0.929	0.925	0.894	0.610	0.670	0.836
Off-Policy Distill.	4.882	5.026	4.289	4.497	4.679	3.797	4.528	0.975	0.919	0.887	0.894	0.580	0.650	0.818
DiffusionOPD [53]	5.488	5.850	4.242	4.303	4.588	5.211	4.947	1.000	0.929	0.975	0.894	0.590	0.610	0.833
Flow-OPD [24]	6.014	5.214	4.467	3.957	4.793	4.681	4.854	0.975	0.909	0.888	0.926	0.550	0.640	0.814
★ DanceOPD (Ours)	5.681	5.857	5.173	5.218	4.840	5.310	5.347	0.988	0.939	0.963	0.894	0.640	0.670	0.849
B. Local and Global Edit Composition														
Joint Training	4.632	5.393	4.128	3.941	4.093	5.086	4.546	0.975	0.939	0.913	0.851	0.600	0.650	0.821
Weight Merge	4.434	4.776	4.380	5.263	5.206	4.229	4.715	0.988	0.909	0.875	0.904	0.600	0.590	0.811
Off-Policy Distill.	5.008	4.683	4.543	4.772	5.075	4.336	4.736	0.975	0.889	0.900	0.872	0.570	0.580	0.798
DiffusionOPD [53]	4.704	5.310	4.502	4.012	4.977	4.462	4.661	0.975	0.899	0.925	0.883	0.620	0.630	0.822
Flow-OPD [24]	4.524	4.647	4.610	5.232	5.037	4.025	4.679	0.975	0.949	0.875	0.872	0.630	0.660	0.827
★ DanceOPD (Ours)	5.178	5.549	6.153	5.944	5.812	4.348	5.498	1.000	0.949	0.925	0.926	0.650	0.640	0.848

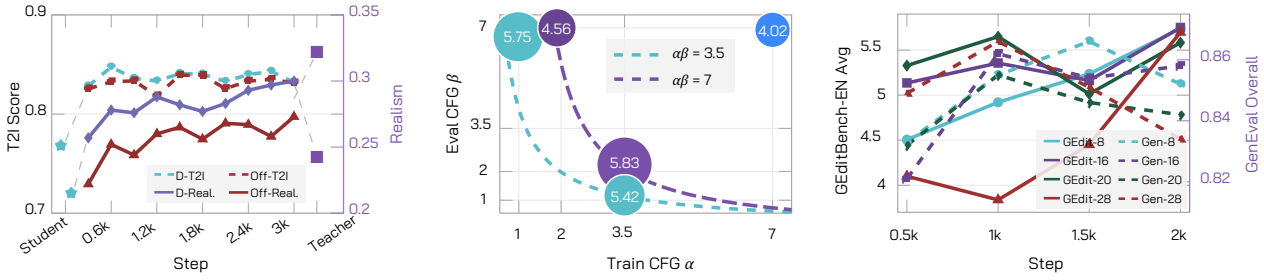

Figure 4 Field Absorption and Rollout Diagnostics. (a) DanceOPD absorbs a realism field more effectively than off-policy distillation while preserving the base text-to-image capability. (b) Training-time and inference-time CFG scales compose, where $\alpha\beta$ denotes the effective CFG in practice. DanceOPD absorbs the training CFG. (c) Moderate rollout discretization suffices for stable field queries during training.

best competing composition baseline in each category, DanceOPD improves background change by 33.5%, style change by 12.9%, and color alteration by 11.6%. This supports the claim that capability-field composition is better handled by routed field queries than by parameter averaging or mixed supervision.

C. Realism Absorption. We next evaluate whether DanceOPD can absorb a realism-oriented quality field that shifts generations toward more photorealistic texture, lighting, and visual statistics while preserving the original T2I capability. DanceOPD improves the realism reward over off-policy distillation by 9.9% and closes 85.3% of the student-to-teacher reward gap. At the same time, its T2I score improves over the student anchor by 7.6% and matches off-policy distillation within 0.1%, indicating that the student moves toward the realism teacher without sacrificing base generation behavior. The qualitative results in Fig. 7 show the same trend. In the ticket-booth example, our model enhances lighting contrast, structural details, and photographic sharpness while keeping the original scene layout. In the portrait and face examples, it improves skin texture, illumination, and water-reflection details without changing the main subject. In the tea-set and leather-shoe examples, DanceOPD produces more coherent material appearance and realistic surface reflections, whereas off-policy distillation is more prone to washed-out structure or over-saturated texture. These results suggest that on-policy querying is useful not only for task composition, but also for absorbing style- or quality-oriented fields while preserving prompt content.

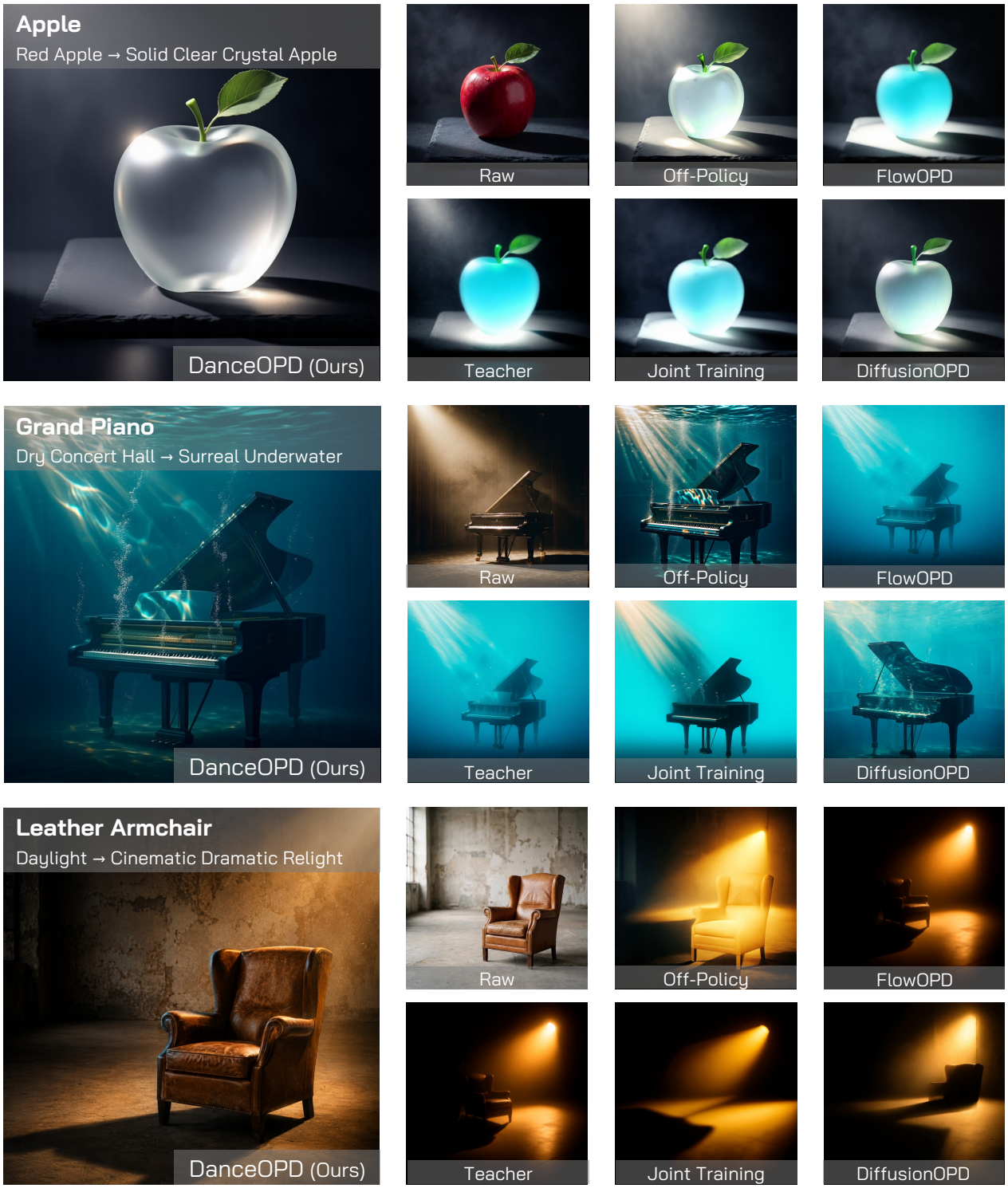

Figure 5 T2I and Edit Composition: Additional Edit Cases. DanceOPD better balances target attribute changes with content preservation across material, lighting, and style edits.

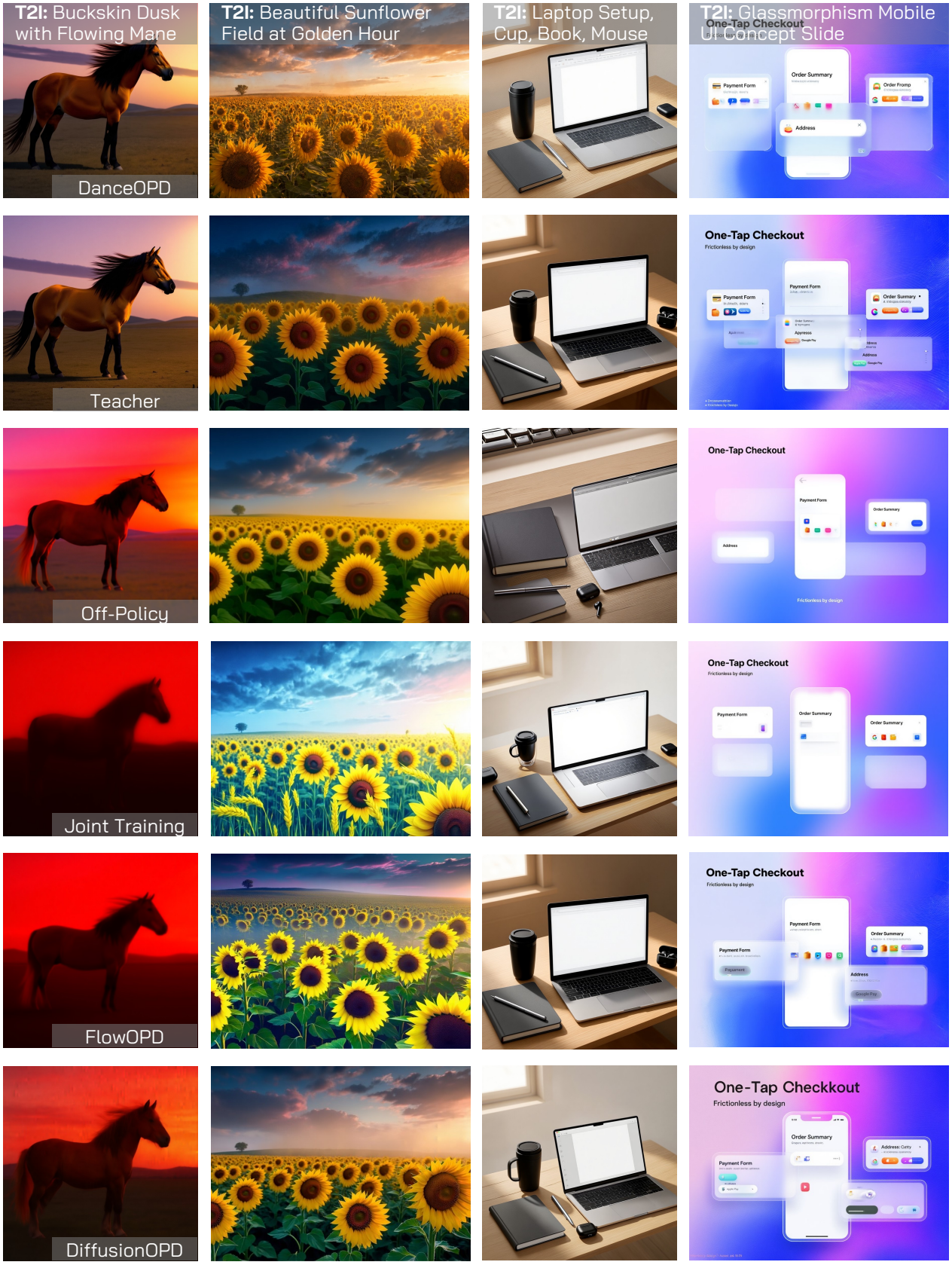

Figure 6 T2I and Edit Composition: Text-to-Image Ability. DanceOPD retains strong text-to-image generation quality while learning editing capabilities, whereas several baselines introduce color shift, blur, or layout degradation.

D. CFG Absorption. We finally evaluate whether DanceOPD can absorb an operator-defined classifier-free guidance field into the student, so that part of the guidance effect is internalized rather than applied only at inference time [36]. We instantiate this case by treating a CFG-guided velocity field as the frozen target field and training the student to approximate it with a single forward pass. This setting also exposes an evaluation caveat: absorbed guidance and external inference-time guidance are not independent. If the absorbed target uses guidance scale α and inference applies scale β again, the effective guidance strength is approximately $\alpha\beta$, so evaluating the absorbed student under the default external-CFG convention can over-guide the model. Fig. 4 (b) confirms this interaction. The best measured composition improves the GEditBench average over train-only absorption by 7.6% and over eval-only CFG by 1.4%. The gains over train-only absorption are especially visible in subject replacement, background change, color alteration, and subject removal, with relative improvements of 10.3%, 22.5%, 6.6%, and 11.2%, respectively. By contrast, excessive composition of absorbed and external guidance reduces the score by 31.2% relative to the best measured composition.

4.2 Multi-Teacher Extension

After the main composition results, we study how several capability fields should supervise the same student during one training process. This setting is related to multi-teacher distillation and task-customized knowledge amalgamation [35, 88], but the key issue here is the field query itself. When several fields enter the same update, the student can lose sample-level semantic identity through soft teacher mixing, or overuse correlated states through dense rollout supervision. We therefore use this subsection to isolate target-field ambiguity and trajectory-query correlation, the two query-level failure modes introduced in Sec. 3.2 and Sec. 3.4.

The diagnostics vary three existing choices while keeping the same capability fields. The first comparison tests hard routing against soft all-teacher mixing. The second tests step alternation against same-step accumulation, where G denotes the number of capability buckets accumulated before one optimizer step. The third varies dense trajectory supervision, where K denotes the number of queried states per rollout, and uses SDE rollout only as a diagnostic control. Table 7 reports the full per-category values.

Hard Routing versus Soft Teacher Mixing. Hard routing is needed because each training sample should be supervised by the capability field that matches its semantic role. Soft all-teacher mixing removes this sample-level identity by averaging all teacher fields into one target, so the target no longer cleanly represents the intended T2I, local edit, or global edit query. This difference is consistent across objectives. With MSE, hard routing improves the average over soft mixing by 15.2%, with gains on subject addition, background change, style change, color alteration, and subject removal. The largest gains appear on background change and subject removal, where hard routing improves by 20.8% and 26.8%, respectively. With KL-$\bar{\sigma}^2$, hard routing still improves the average over soft mixing by 10.6%, including 14.5% on style change and 14.4% on subject removal. These results show that the main issue is target-field construction, rather than only the choice of matching objective.

Same Step Multi-Teacher Accumulation. Step alternation keeps each optimizer update tied to one routed capability bucket. Same-step accumulation keeps the routed target for each individual sample, but combines several capability buckets before one optimizer step. This tests whether routing remains sufficient when multiple capability fields affect the same update. Compared with the single-query step-alternation default, same-step accumulation with $K{=}1$ and $G{=}3$ loses 4.6% on average. The change is not a uniform degradation across all categories. Style change and color alteration improve, but subject addition and subject removal drop by 17.5% and 13.5%, showing that same-step accumulation shifts the balance between capabilities rather than preserving all of them. When dense supervision is also added, the $K{=}2, G{=}3$ case loses 22.8% on average, with subject addition and subject removal dropping by 28.9% and 46.0%. This shows that capability conflict can reappear at the optimizer-update level, even when every individual sample keeps a valid routed target [56, 107].

Dense Query Drift Control. The dense-query diagnostic tests whether the poor dense same-step result is tied to trajectory-query correlation. In the $K{=}2, G{=}3$ stress case, replacing ODE rollout with SDE rollout improves the average by 18.4%. The recovery is strongest on subject removal and subject addition, which improve by 62.0% and 29.0%, and it also improves subject replacement and background change by 15.4% and 10.8%. However, the rescued result still remains 8.6% below the single-query default, and it is weaker on style change,

Figure 7 Realism-Field Absorption. DanceOPD absorbs the teacher's realism-oriented field more effectively than off-policy distillation, shifting samples toward more photorealistic texture, lighting, and visual statistics while preserving the base model's prompt-following and content generation ability.

color alteration, and subject removal by 18.7%, 9.8%, and 12.5%. SDE rollout also hurts the two control settings by 15.3% and 8.3%. Thus, stochastic rollout noise is useful as diagnostic evidence for trajectory-query correlation, but it is not a better default. The safer design is to avoid correlated dense supervision in the first place and use a single semantic-side query.

4.3 Ablation Study

The previous subsection studied how multiple capability fields should enter the same student update. We now ablate the remaining training choices of DanceOPD under the same frozen capability fields and the same evaluation protocol. These choices include the number of rollout steps used to obtain student states, the timestep region where the teacher field is queried, the number of queried states from one rollout, the local matching objective, and the student initialization. Fig. 9 summarizes the main trends, while Table 8 and Table 9 report the detailed values.

Rollout Steps. The rollout is used to obtain student-visited states for teacher queries, which generally need to be consistent with inference. However, we found that in DanceOPD, the training rollout length does not need to match the evaluation sampler exactly. This may be because our solver is ODE [47, 62, 90]. In the DanceOPD experiments, at 2000 steps, a moderate rollout length is sufficient. At 2k steps, the 16-step rollout gives the strongest GEditBench average, improving over the 8-step, 20-step, and 28-step variants by 0.2%, 3.0%, and 0.9%, respectively. It also improves GenEval overall over these variants by 0.7%, 1.9%, and 2.9%. Although the 28-step rollout is competitive on some edit sub-scores, the 16-step rollout improves subject removal over it by 33.7% and gives better T2I preservation. This shows that longer rollout discretization does not automatically provide better supervision for field matching.

Timestep Query. Capability-specific supervision is most useful on the semantic, low-noise side of the trajectory,

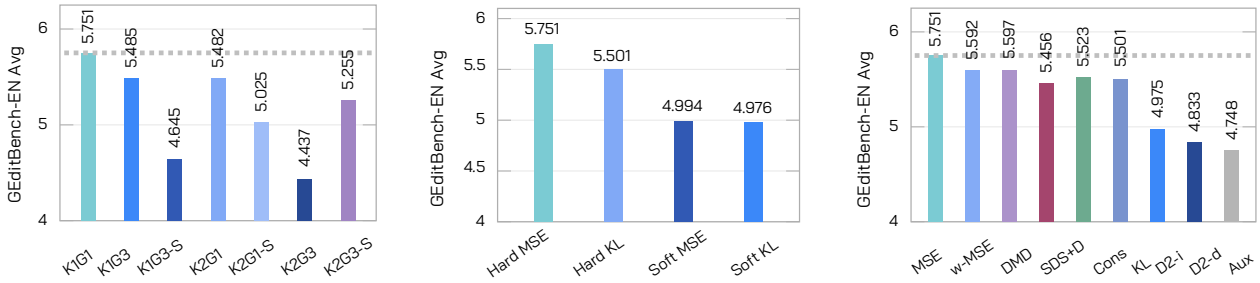

Figure 8 Routing, Objective, and Dense-Query Diagnostics. Hard routing with single-query MSE is the strongest default; same-step accumulation, soft teacher mixing, and dense correlated queries degrade performance.

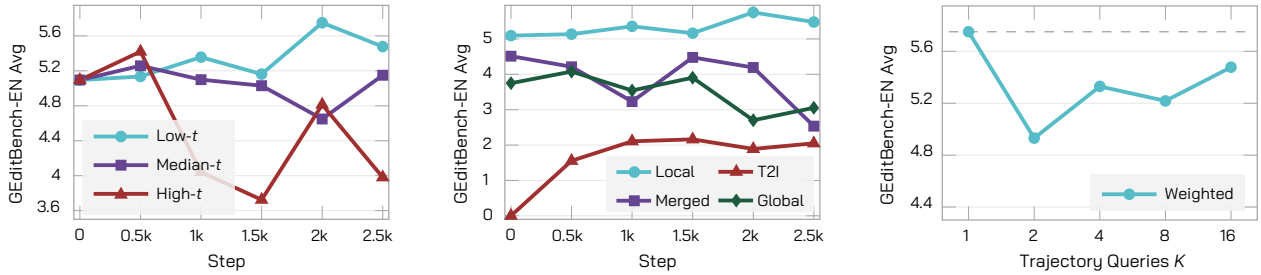

Figure 9 Ablation Trends. Semantic-side low-t queries and local-edit initialization give the most reliable gains, while increasing the number of trajectory queries does not improve performance.

consistent with timestep-sensitive image editing and distillation analyses [11, 54, 96]. At 2k steps, low-t querying improves the GEditBench average over median-t and high-t querying by 23.7% and 19.5%. Compared with median-t, low-t improves subject addition by 35.9%, background change by 36.1%, color alteration by 14.0%, and subject removal by 42.3%. Compared with high-t, low-t improves subject addition by 46.1% and style change by 57.1%. These results support the design choice of querying the teacher field on the semantic side of the student rollout.

Number of Trajectory Queries. Using more states from the same rollout does not necessarily improve supervision, because states along one rollout are highly correlated, echoing recent observations that rollout position and rollout length affect OPD stability [111]. The single-query default outperforms the weighted dense-query variants across $K{=}2, 4, 8, 16$, improving the GEditBench average by 16.6%, 7.9%, 10.2%, and 12.2%, respectively. Even the strongest weighted dense-query variant remains below the single-query default. This confirms that one semantic-side query per rollout is not only more efficient, but also more reliable for matching the routed teacher field.

Objective Design. Plain velocity MSE gives the strongest average among the tested local matching objectives. It improves over timestep-weighted MSE and DMD-EMA hybrid by 2.8%, over consistency matching by 4.1%, and over KL-$\bar{\sigma}^2$ matching by 4.5%. The gap is larger for DMD2 and AuxFeat variants, where plain velocity MSE improves the average by 15.6% to 21.1%. Some alternatives improve individual sub-scores, but they do not give the same overall balance across edit categories. Since the target is a routed velocity field queried at a fixed student state, direct velocity regression gives the most stable update.

Initialization. Initialization matters because the student rollout determines where teacher fields are queried early in training. The best default is the checkpoint with the strongest relevant capability, rather than merged initialization or an unrelated anchor. At 2k steps, local edit initialization improves the GEditBench average over merged initialization by 37.2%, over global edit initialization by 112.8%, and over T2I initialization by 204.4%. Compared with merged initialization, local edit initialization improves subject addition by 48.8%, subject replacement by 32.1%, background change by 28.7%, color alteration by 29.2%, and subject removal by 124.0%. This indicates that a stronger initial student rollout makes field matching more effective.

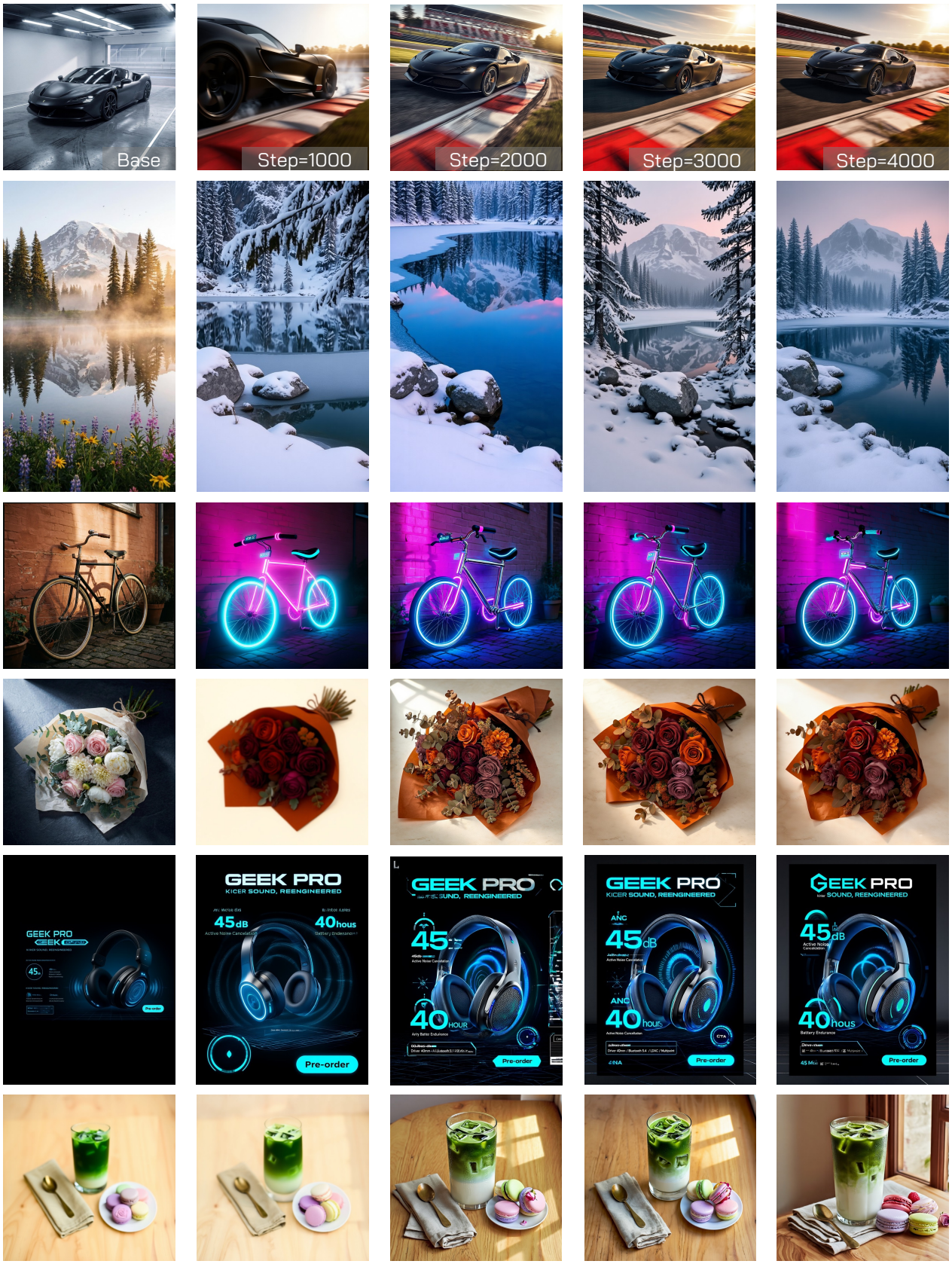

Figure 10 Training Progression in Local and Global Edit Composition. As distillation proceeds, DanceOPD progressively absorbs the target edit capability while maintaining scene identity and prompt alignment.

4.4 Capabilities Emergence

An unexpected qualitative finding arises when we evaluate zero-shot tasks that are absent from the training distribution. Using the final checkpoint from the T2I and Edit Composition setting in Table 2, we test reference-conditioned generation tasks for which neither source model was explicitly trained. As shown in Fig. 11, DanceOPD is the only evaluated method that consistently exhibits capabilities unavailable to both the T2I teacher and the edit-initialized student. The T2I teacher is the text-to-image Z-Image model, which cannot interpret the additional input image and therefore produces Gaussian-noise outputs under the editing interface. The student is initialized from a Z-Image Edit model trained on OmniEdit. However, OmniEdit contains only local and global editing data and does not include the novel reference-generation tasks considered here, such as multi-shot generation, viewpoint changes, or outfit decomposition.

In outfit decomposition [Fig. 11(a)], DanceOPD separates the clothing into semantically meaningful individual garments rather than copying the complete input. In multi-shot generation [Fig. 11(b)], it preserves the identity of the subject while producing subsequent views with clear changes in shot and viewpoint. Flow-OPD, by contrast, remains substantially closer to the structure of the source image. This behavior indicates that DanceOPD does not merely select or reproduce the output of one teacher. Because the distinct capability fields remain well separated within a shared student, the model can recombine existing behaviors and exhibit stronger capabilities when a test request lies between the original training tasks.

This result is also notable from the perspective of timestep specialization. Under the conventional coarse-to-fine interpretation of diffusion trajectories, high-noise, high-t states are primarily responsible for establishing global composition and spatial layout, whereas low-noise, low-t states mainly refine local appearance and details and are therefore not expected to reconstruct a substantially different layout [96]. DanceOPD, however, is supervised through only one low-t query, yet its outputs on these held-out tasks differ qualitatively from those of both the T2I teacher and the edit-initialized student. The model therefore does more than copy the layout of either endpoint; it appears to synthesize a new spatial organization from the composed capability fields. This departure from the usual timestep interpretation provides another intriguing indication of emergent behavior.

Flow-OPD and DiffusionOPD instead tend to copy the behavior associated with the editing branch. We hypothesize that the additional image input makes these requests more likely to be treated as editing tasks, causing their outputs to remain closer to the edit-initialized student, while Off-Policy Distillation and Joint Training show inconsistent behavior. We further conjecture that the stronger zero-shot generalization of DanceOPD is related to its single-query design. By avoiding dense, correlated supervision along the same rollout, single-query field matching may preserve more clearly separated and recomposable capability fields, enabling novel behaviors to emerge on previously unseen tasks.

5 Conclusion

We introduced DanceOPD, an on-policy generative field distillation framework for composing heterogeneous generative capabilities into a single flow-matching student. The key view is to treat each frozen capability source as a velocity field over a shared generative state space and to formulate capability composition as a field-query problem. This perspective exposes three query-induced alignment challenges, including target-field ambiguity, state-distribution mismatch, and trajectory-query correlation. Correspondingly, our model uses hard-routed sample-wise field matching to preserve the semantic identity of each capability, queries the routed field on student-visited states from the current rollout, and uses one semantic-side low-noise query per sample to avoid correlated within-trajectory supervision. With a plain velocity MSE objective, this design composes and absorbs capability fields without relying on joint training, parameter-space merging, or external score composition.

Experiments on T2I and editing composition, local and global editing composition, realism-field absorption, and CFG absorption show that DanceOPD strengthens the target capability while preserving the anchor capability. Ablations further support the role of hard routing, student-induced query states, semantic-side single-query supervision, and direct velocity regression. These results suggest that on-policy generative field distillation is a practical route toward scalable multi-capability visual generation.

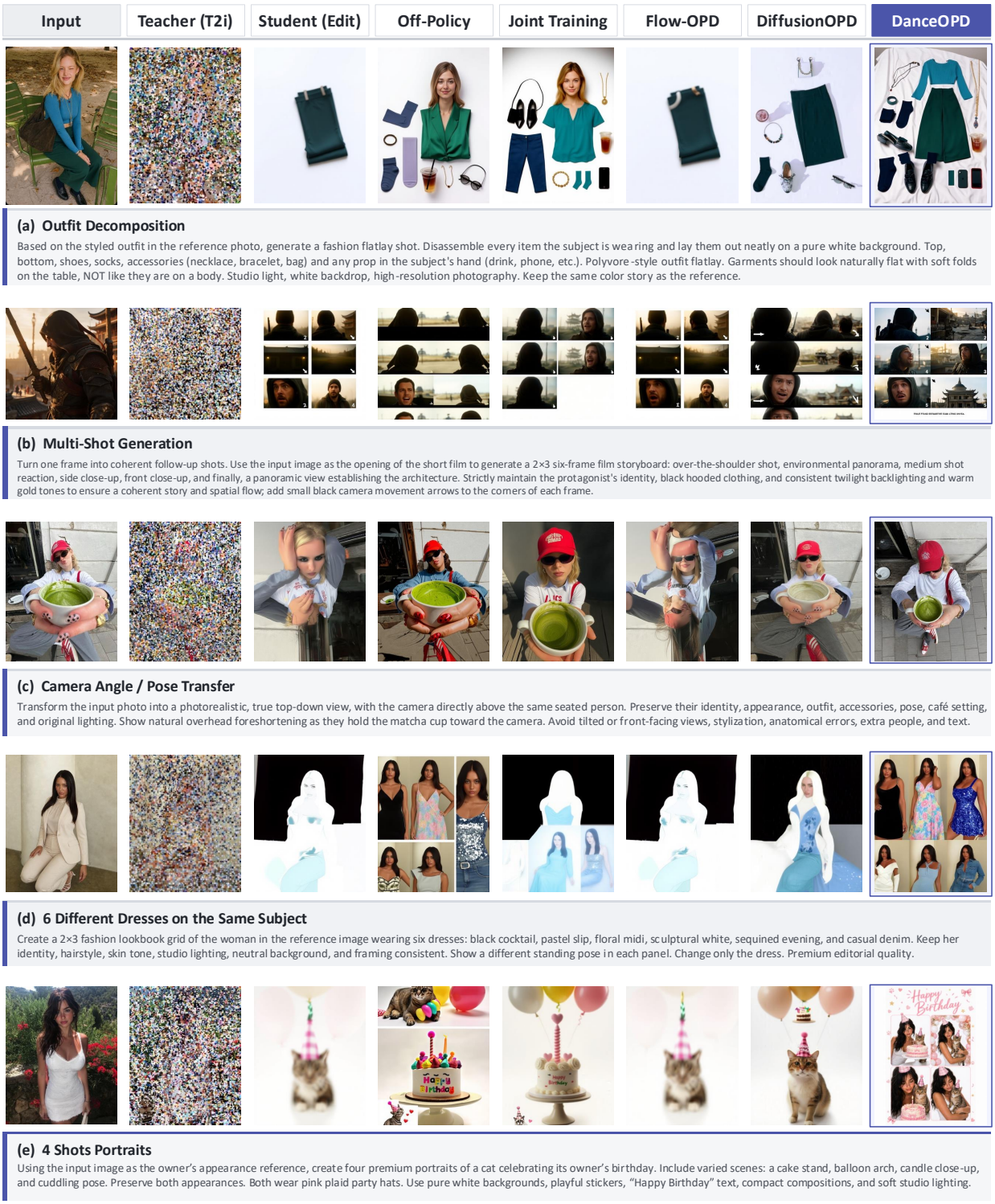

Figure 11 Zero-Shot Capability Emergence on Unseen Reference-Conditioned Tasks. Neither the T2I teacher nor the edit-initialized student is explicitly trained for these tasks. DanceOPD nevertheless exhibits coherent outfit decomposition, identity-preserving multi-shot generation, camera-angle and pose transfer, and reference-based visual recomposition, while the competing methods all fail.

6 Limitations and Discussions

Shared Field Support. The current formulation assumes that frozen capability sources expose compatible velocity fields over a shared generative state space. In our experiments, this assumption is satisfied because the sources are built from the same backbone family, latent representation, scheduler convention, and velocity parameterization. This requirement is not unique to DanceOPD: LLM OPD methods usually require teacher and student distributions to be defined over the same token space before token-level KL or asymmetric KL can be applied [1, 30, 45], and flow-matching OPD methods also require compatible transition, score, or velocity parameterizations for teacher and student matching [24, 46, 53].

Predefined Routing. Our implementation uses predefined capability buckets and sample-wise hard routing. This is a standard and stable OPD-style design when the supervision source is known from the task or data identity, and it is well matched to our setting, where T2I, local editing, global editing, style, and guidance-field examples are easy to separate. However, this assumption becomes weaker when task boundaries are ambiguous or when a prompt requires several capabilities at once. A natural extension is to introduce a verifier or reward model that assigns routes according to model behavior or predicted edit success, connecting our field-routed distillation to reward- and verifier-based evaluation or supervision signals [23, 49, 51, 99, 100].

7 Theoretical Details

This appendix provides the theoretical motivation behind DanceOPD and follows the same field-query logic as the main text. Given frozen capability fields over a shared flow state space, the student must decide which field to query, which student-induced state to query, and how many states from the same rollout to use.

The local analyses below explain why plain velocity MSE is a natural field-matching objective, why teacher fields should be queried on stop-gradient states from the current student rollout, why mixing several fields inside one sample target creates target-field ambiguity, and why dense same-rollout queries can suffer from trajectory-query correlation. The experiments then test these predicted failure modes and design choices in full image-generation models.

7.1 KL–MSE Equivalence for Velocity-Field Distillation

DanceOPD uses plain velocity MSE in the main method. Here, we show why this objective is the natural form of a KL-style distillation loss when both the student and frozen capability source define local Gaussian transition kernels over the same flow state space.

Consider a small reverse-time step from t to $t - \Delta t$. Let the student and the routed capability field induce local Gaussian kernels with shared covariance $\sigma_t^2 I$:

$$p_\theta(z_{t-\Delta t} \mid z_t, c) = \mathcal{N}\big(z_t - \Delta t\, v_\theta(z_t, t, c),\, \sigma_t^2 I\big), \tag{8}$$

$$p_m(z_{t-\Delta t} \mid z_t, c) = \mathcal{N}\big(z_t - \Delta t\, v_m(z_t, t, c),\, \sigma_t^2 I\big). \tag{9}$$

For two Gaussians with identical covariance, the forward KL has a closed form:

$$D_{\mathrm{KL}}(p_m \,\|\, p_\theta) = \frac{\Delta t^2}{2\sigma_t^2} \left\| v_\theta(z_t, t, c) - v_m(z_t, t, c) \right\|_2^2 . \tag{10}$$

The reverse KL has the same quadratic form under the same covariance assumption. Thus, KL-style local transition matching reduces to a timestep-weighted velocity MSE. This is consistent with flow matching, which directly regresses vector fields along a probability path [55], and with diffusion and SDE formulations where Gaussian perturbation kernels yield quadratic score or velocity matching objectives [19, 37, 71, 89–92]. In the main method, we use the unweighted MSE because the teacher target is deterministic, and our ablations find the plain objective more stable than the tested weighting variants.

7.2 On-Policy Query Distribution and Stop-Gradient Rollout

Let Φ_θ^t denote the numerical solver that maps an initial noise z_T to an intermediate state at time t using the current student velocity, and let p_T denote the initial-noise distribution. DanceOPD samples a semantic-side coordinate and maps it to a physical timestep before querying the rollout:

$$s \sim q_{\text{sem}}(s), \quad t = t(s), \quad z_t^\theta = \Phi_\theta^t(z_T, c), \qquad z_T \sim p_T. \tag{11}$$

Conditioned on a routed sample $m \sim \pi$ and $(x, c) \sim \mathcal{D}_m$, the implemented loss uses the sampled state as a query point, that is:

$$\mathcal{L}_{\text{sg}} = \mathbb{E}_{m\sim\pi,\,(x,c)\sim\mathcal{D}_m,\,z_T\sim p_T,\,s\sim q_{\text{sem}}} \left[\left\| v_\theta(\text{sg}(z_t^\theta), t, c) - v_m(\text{sg}(z_t^\theta), t, c) \right\|_2^2 \right], \qquad t = t(s), \tag{12}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. Thus, the gradient is:

$$\nabla_\theta \mathcal{L}_{\text{sg}} = 2\,\mathbb{E}_{m,(x,c),z_T,s} \left[\left(v_\theta - v_m\right)^\top \nabla_\theta v_\theta(\text{sg}(z_t^\theta), t, c) \right], \tag{13}$$

where v_θ and v_m inside the inner product are evaluated at $(\text{sg}(z_t^\theta), t, c)$, with the capability target detached. This is the practical first-order estimator: it avoids differentiating through every solver step while still querying the field on the student's current state distribution. This corresponds to the second query choice in the main text, where the teacher field is evaluated on the states visited by the current student rather than on fixed off-policy states.

The reason to query on-policy can be stated as a simple mismatch bound. Suppose an off-policy method queries a state $\tilde{z}_t$ while the deployed student visits z_t^θ. If the routed capability field is L_m-Lipschitz in z, then

$$\left\| v_m(z_t^\theta, t, c) - v_m(\tilde{z}_t, t, c) \right\|_2 \le L_m \left\| z_t^\theta - \tilde{z}_t \right\|_2. \tag{14}$$

Therefore, teacher supervision collected far from the student rollout can be a biased local target for deployment-time states. This mirrors the covariate-shift motivation behind dataset aggregation in imitation learning [79], but here the states are continuous diffusion and flow trajectories rather than discrete actions.

7.3 Smoothness of Capability Fields on Student-Rolled States

A natural concern is that a student rollout may visit states different from those produced by a frozen capability source. The following standard flow-matching view explains why this does not automatically invalidate the queried target.

For a rectified-flow interpolant with data endpoint x_0 and noise endpoint x_1, the intermediate state can be written as:

$$z_t = (1 - t)x_0 + t x_1. \tag{15}$$

The optimal flow-matching velocity is a conditional expectation [55], that is:

$$v^\star(z, t) = \mathbb{E}[x_1 - x_0 \mid z_t = z]. \tag{16}$$

Conditional expectations over Gaussian-convolved marginals vary smoothly with z away from degenerate endpoints. Let v_m be a trained capability field whose approximation error to its ideal field is bounded by δ_m on the relevant marginal support, and let the ideal field be L_t-Lipschitz in z. For a student-visited state z_t^θ and a nearby reference state $\tilde{z}_t$ satisfying $\|z_t^\theta - \tilde{z}_t\|_2 \le \varepsilon$, we have:

$$\left\| v_m(z_t^\theta, t, c) - v_m(\tilde{z}_t, t, c) \right\|_2 \le L_t \varepsilon + 2\delta_m. \tag{17}$$

The bound follows by adding and subtracting the ideal field and applying the triangle inequality. Its role is not to prove that arbitrary off-policy states are safe; rather, it formalizes the regime used by DanceOPD: when the current student remains within a bounded neighborhood of the broad flow marginal, the frozen field gives an interpolative and locally meaningful target. This is why the main method can query fields on student-rolled states, while dense-query variants still require correlation analysis.

7.4 Hard-Routed Estimator and Field-Conflict Bias

The hard-routed objective in Sec. 3.2 can be written with $\pi_m = \pi(m)$ as follows:

$$\mathcal{L}_{\text{route}} = \sum_{m=1}^{M} \pi_m \, \mathbb{E}_{(x,c)\sim\mathcal{D}_m,\, z_T\sim p_T,\, s\sim q_{\text{sem}}} \ell_m(x, c, z_T, s), \qquad \ell_m(x, c, z_T, s) = \left\| v_\theta(z_t^\theta, t, c) - v_m(z_t^\theta, t, c) \right\|_2^2, \quad t = t(s). \tag{18}$$

Here $z_t^\theta = \Phi_\theta^{t(s)}(z_T, c)$ is the stop-gradient student-rolled query state. Sampling one route $m \sim \pi$ and one sample from $\mathcal{D}_m$ gives an unbiased stochastic estimator of Eq. (18). This is the reason the main algorithm does not need to query all fields at every step: the long-run optimization still covers all capabilities, while each individual sample keeps a single semantic target. This corresponds to the first query choice in the main text. The route decides which frozen capability field supervises the current sample, so the target remains a well-defined capability query rather than an average of several fields.

Let y denote the route label of the current sample. A within-sample mixture with nonnegative weights w_m, in contrast, replaces the routed target v_y by:

$$\bar{v} = \sum_{m=1}^{M} w_m v_m, \qquad w_m \geq 0, \quad \sum_m w_m = 1. \tag{19}$$

The target bias relative to the correct field is:

$$\bar{v} - v_y = \sum_{m \neq y} w_m \, (v_m - v_y). \tag{20}$$

When non-route fields encode unrelated or conflicting capabilities, Eq. (20) injects irrelevant directions into the sample. This is the algebraic form of target-field ambiguity. Even if the mixed target is numerically valid, its direction need not correspond to the capability associated with the current sample.

Same-step accumulation avoids this target bias but still averages gradients from G capability buckets in one optimizer update. Let g_r denote the gradient from the r-th routed bucket; then:

$$g_{\text{acc}} = \frac{1}{G} \sum_{r=1}^{G} g_r, \qquad \|g_{\text{acc}}\|^2 = \frac{1}{G^2} \sum_{i=1}^{G} \sum_{j=1}^{G} \langle g_i, g_j \rangle. \tag{21}$$

If cross-field gradient inner products are small or negative, the same-step update becomes a compromise direction. Here G matches the capability-bucket count used in the diagnostic tables. This is the mathematical form of the capability-field conflict tested in Sec. 4.2, and is the same phenomenon targeted by gradient-balancing, multi-objective, gradient surgery, and conflict-averse multi-task optimization methods [13, 44, 56, 85, 107].

7.5 Semantic-Side Query Distribution and Dense Aggregation

Let $s \in [0, 1]$ denote a semantic-side coordinate, with larger s closer to the low-noise, clean-image side of the trajectory. The main method samples one query with positive Beta parameters α_{sem} and β_{sem}:

$$s \sim \text{Beta}(\alpha_{\text{sem}}, \beta_{\text{sem}}), \qquad \alpha_{\text{sem}} > \beta_{\text{sem}}, \tag{22}$$

then maps s to the corresponding solver state z_t^θ. Median- and high-noise query ablations are obtained by changing the Beta parameters or using a uniform distribution. The role of Eq. (22) is not to tune a sampler arbitrarily, but to query a region where capability-specific information is dense. This gives the implementation form of the semantic-side single query used in the main method. The query is placed in the low-noise region where edit, style, and visual-attribute information is more concentrated, while still using the current student rollout state.

Table 3 Timestep-Query Distributions. The rollout index increases from the high-noise beginning of the reverse trajectory to the low-noise semantic side. Therefore, a distribution with a larger mean in s selects later rollout states, which correspond to lower physical noise time.

Ablation Label	Distribution Over s	Mean Of s	Selected Rollout Region	Interpretation
Low-t semantic-side	Beta$(5,2)$	$\frac{5}{5+2} \approx 0.714$	late indices	low physical t, near-clean, semantic-rich
Median-t	Beta$(5,5)$	0.5	middle indices	intermediate-noise query
High-t noise-side	Beta$(2,5)$	$\frac{2}{2+5} \approx 0.286$	early indices	high physical t, noise-side query

Table 3 spells out the convention used in the ablation table: the Beta random variable is sampled over the normalized rollout index, or semantic coordinate s, not directly over the physical flow time t.

For dense-query diagnostics with $K > 1$, the deterministic ODE rows use the student rollout:

$$z_{i+1} = z_i - \Delta t\, v_\theta(z_i, t_i, c), \tag{23}$$

and query K states $\{z^\theta_{t_k}\}_{k=1}^K$ from this trajectory. Let:

$$\ell_k = \left\| v_\theta(z^\theta_{t_k}, t_k, c) - v_m(z^\theta_{t_k}, t_k, c) \right\|_2^2 \tag{24}$$

be the routed velocity loss at the k-th queried state. We consider generic aggregations, where a_k denotes weighted-aggregation coefficients, d_k denotes a detached loss value, and $\epsilon_{\text{stab}} > 0$ is a numerical stabilizer:

$$\mathcal{L}_{\text{mean}} = \frac{1}{K} \sum_{k=1}^K \ell_k, \tag{25}$$

$$\mathcal{L}_{\text{sum}} = \sum_{k=1}^K \ell_k, \tag{26}$$

$$\mathcal{L}_{\text{wtd}} = \sum_{k=1}^K a_k \ell_k, \tag{27}$$

$$a_k = \frac{d_k}{\sum_{j=1}^K d_j + \epsilon_{\text{stab}}}, \tag{28}$$

$$d_k = \text{sg}(\ell_k), \tag{29}$$

Here $a_k \geq 0$ and $\sum_k a_k \approx 1$. Mean aggregation keeps the loss scale fixed but dilutes each state; sum aggregation preserves per-state gradient magnitude but is more sensitive to correlated residuals; weighted aggregation gives higher-disagreement states more credit while avoiding an uncontrolled raw sum.

In Fig. 8 (a), "isolation" denotes a G=1 hard-routing control that increases K without same-step multi-teacher accumulation, while "sum no control" denotes Eq. (24) aggregated by $\mathcal{L}_{\text{sum}}$ under ODE rollout, without weighting or SDE decorrelation. These are implementation choices for diagnostics, not the default method.

7.6 Loss Variants Considered

Let v_m denote the hard-routed teacher velocity for the current sample, and let z^θ_t be the stop-gradient student-rolled query state. Throughout this subsection, unadorned velocity terms are evaluated at (z^θ_t, t, c) unless their arguments are shown explicitly. The default objective is direct velocity regression, that is:

$$\mathcal{L}_{\text{MSE}} = \left\| v_\theta(z^\theta_t, t, c) - v_m(z^\theta_t, t, c) \right\|_2^2. \tag{30}$$

The objective ablations in Fig. 8 (c) use the same K=1, G=1 single-query ODE setting unless the row explicitly changes the teacher schedule or dense-query setting.

Timestep-Weighted and KL-Weighted MSE. We test two weighted velocity-regression forms, using a Min-SNR-style weight $w_{\mathrm{SNR}}(t)$ and local transition variance $\bar{\sigma}_t^2$:

$$\mathcal{L}_{\mathrm{SNR}} = w_{\mathrm{SNR}}(t)\,\|v_\theta - v_m\|_2^2\,, \tag{31}$$

$$\mathcal{L}_{\mathrm{KL}\text{-}\bar{\sigma}^2} = \frac{\Delta t^2}{2\bar{\sigma}_t^2}\,\|v_\theta - v_m\|_2^2\,. \tag{32}$$

The first objective uses timestep weighting; the second follows the Gaussian local-transition KL view in Sec. 7.1. Both keep the same deterministic teacher target as Eq. (30).

DMD-EMA Hybrid and Pure DMD. Let v_{ema} be a passive EMA reference velocity field, and let $\beta \in [0,1]$ be the MSE-anchor weight. The DMD-EMA hybrid uses a score-style surrogate plus an MSE anchor, that is:

$$\mathcal{L}_{\mathrm{DMD\text{-}EMA}} = -(1-\beta)\left\langle \big(v_m - v_{\mathrm{ema}}\big)_{\mathrm{sg}}, v_\theta \right\rangle + \beta\,\|v_\theta - v_m\|_2^2\,. \tag{33}$$

Pure DMD is the special case $\beta = 0$:

$$\mathcal{L}_{\mathrm{pure\text{-}DMD}} = -\left\langle \big(v_m - v_{\mathrm{ema}}\big)_{\mathrm{sg}}, v_\theta \right\rangle. \tag{34}$$

This removes the velocity-regression anchor and is therefore a stability diagnostic rather than a proposed replacement.

SDS+DMD-EMA Combined. The combined score-style row applies timestep weighting to the score surrogate and keeps the MSE anchor, that is:

$$\mathcal{L}_{\mathrm{SDS+DMD}} = -(1-\beta)\,w_{\mathrm{SNR}}(t)\left\langle \big(v_m - v_{\mathrm{ema}}\big)_{\mathrm{sg}}, v_\theta \right\rangle + \beta\,\|v_\theta - v_m\|_2^2\,. \tag{35}$$

It tests whether timestep weighting and a score-style direction are complementary.

DMD2-Inter and DMD2-Dist. The DMD2 rows replace the passive EMA field with a trained fake reference critic v_{fake} and use mixing weight $\beta_2 \in [0,1]$, that is:

$$\mathcal{L}_{\mathrm{DMD2}} = -(1-\beta_2)\left\langle \big(v_m - v_{\mathrm{fake}}\big)_{\mathrm{sg}}, v_\theta \right\rangle + \beta_2\,\|v_\theta - v_m\|_2^2\,. \tag{36}$$

The DMD2-inter and DMD2-dist rows in Fig. 8 (c) are implementation variants of this same score-surrogate family inspired by adversarial and distribution matching distillation [82, 83, 104]: DMD2-inter trains the fake critic on rollout-intermediate targets, whereas DMD2-dist trains it on fresh student-distribution denoising targets.

Consistency Variant. For two nearby queried states z_{t_i} and z_{t_j}, we form a linear clean-image estimate $\hat{x}_{0,a}(z_t) = z_t - t\,v_a(z_t,t,c)$ for $a \in \{\theta, m\}$ and use nonnegative weights λ_{self} and $\lambda_{\mathrm{anchor}}$:

$$\mathcal{L}_{\mathrm{cons}} = \lambda_{\mathrm{self}}\left\|\hat{x}_{0,\theta}(z_{t_i}) - \mathrm{sg}\big(\hat{x}_{0,\theta}(z_{t_j})\big)\right\|_2^2 + \lambda_{\mathrm{anchor}}\,\|\hat{x}_{0,\theta}(z_{t_i}) - \hat{x}_{0,m}(z_{t_i})\|_2^2\,. \tag{37}$$

This is a simplified consistency regularizer [48, 93] with a teacher anchor.

Auxiliary Feature Distillation. Let h_θ^ℓ and h_m^ℓ be matched student and teacher hidden features at selected DiT layers $\ell \in \mathcal{S}$, and let $\lambda_{\mathrm{feat}} \geq 0$ be the feature-distillation weight, following the broader practice of hidden-feature, attention, and representation-level distillation [2, 32, 78, 108]. The AuxFeat row adds a feature-level term to the velocity MSE, that is:

$$\mathcal{L}_{\mathrm{AuxFeat}} = \|v_\theta - v_m\|_2^2 + \lambda_{\mathrm{feat}}\frac{1}{|\mathcal{S}|}\sum_{\ell\in\mathcal{S}}\left\|h_\theta^\ell - \mathrm{sg}(h_m^\ell)\right\|_2^2. \tag{38}$$

Soft-Teacher Mixing Objectives. The hard-routed KL-$\bar{\sigma}^2$ row and the soft-teacher rows change different axes. KL-$\bar{\sigma}^2$ keeps sample-wise routing and changes only the timestep weighting of a single routed teacher

target. Soft-teacher mixing removes routing: for the same student-rolled state, all capability fields are queried and combined inside the same sample objective, similar in spirit to multi-teacher aggregation but without sample-level task selection [88].

The soft-teacher MSE row uses nonnegative mixture weights w_m:

$$\mathcal{L}_{\text{soft-MSE}} = \sum_{m=1}^{M} w_m \left\| v_\theta(z_t^\theta, t, c) - v_m(z_t^\theta, t, c) \right\|_2^2, \qquad w_m \geq 0, \quad \sum_{m=1}^{M} w_m = 1. \tag{39}$$

Equivalently, with $\bar{v} = \sum_m w_m v_m$, that is:

$$\sum_m w_m \|v_\theta - v_m\|_2^2 = \|v_\theta - \bar{v}\|_2^2 + \sum_m w_m \|v_m - \bar{v}\|_2^2. \tag{40}$$

Therefore, this has the same gradient as MSE to the mixed velocity target $\bar{v}$. It is also a KL-equivalent quadratic under the equal-covariance Gaussian view of Sec. 7.1; the important difference from KL-$\bar{\sigma}^2$ is the non-routed all-teacher target, not a different Gaussian algebra.

The soft-teacher KL-$\bar{\sigma}^2$ diagnostic applies the same local-transition weighting as the hard-routed KL-$\bar{\sigma}^2$ row to the soft all-teacher objective:

$$\mathcal{L}_{\text{soft-KL-}\bar{\sigma}^2} = \frac{\Delta t^2}{2\bar{\sigma}_t^2} \sum_{m=1}^{M} w_m \left\| v_\theta(z_t^\theta, t, c) - v_m(z_t^\theta, t, c) \right\|_2^2. \tag{41}$$

Together, the hard versus soft rows and the MSE versus KL-$\bar{\sigma}^2$ rows form the routing and objective 2×2 summarized in panel b of Fig. 8.

These variants correspond to timestep-weighted KL and MSE matching, SNR-style weighting [31], score-form surrogate gradients inspired by score distillation and distribution matching [75, 82, 83, 104], consistency regularization [48, 93], feature distillation, and soft multi-teacher matching. They are useful ablations, but the unweighted MSE remains our default because it directly matches deterministic capability velocities and empirically gives the strongest stability and performance trade-off.

Implementation Audit. Several alternative losses are useful to test, but should be interpreted carefully in our setting. First, the score-distillation-style variants operate on stop-gradient query states; there is no differentiable rendering or sampling path through which an SDS gradient can act on the query state, unlike reward- or preference-optimized diffusion tuning methods that explicitly optimize through policy-gradient, reward-backpropagation, or preference objectives [7, 22, 76, 95]. Second, the DMD-EMA variant uses a passive EMA reference rather than a separately trained fake-distribution critic. Without the MSE anchor, the score-style gradient need not vanish when the student already matches the routed field, which explains the observed instability of the pure DMD variant. Third, the consistency variant is a simplified diagnostic: it uses a linear $\hat{x}_0$ estimate and a stop-gradient current-student target instead of a full EMA consistency model.

These audits are the reason we present the non-MSE objectives as ablations rather than as replacements for routed velocity regression.

7.7 Guidance-Field Absorption

Classifier-free guidance forms an affine velocity field from the unconditional velocity $v_\emptyset$ and conditional velocity v_{cond} [36]:

$$v_\alpha(z_t, t, c) = v_\emptyset(z_t, t) + \alpha\big(v_{\text{cond}}(z_t, t, c) - v_\emptyset(z_t, t)\big). \tag{42}$$

DanceOPD can treat Eq. (42) as just another capability field and match it by MSE. In the realizable case, the minimizer of the absorbed-CFG loss

$$\mathcal{L}_{\text{CFG}} = \mathbb{E} \left\| v_\theta(z_t^\theta, t, c) - v_\alpha(z_t^\theta, t, c) \right\|_2^2 \tag{43}$$

is $v_\theta = v_\alpha$ on the queried state distribution. The guidance direction is then absorbed into the trained student, so the default deployment can use the distilled field directly rather than recomputing the affine extrapolation.

This also clarifies an implementation pitfall. If a model has already learned an α-guided conditional branch, but inference applies classifier-free guidance with scale β again, then under the approximation $v_{\theta,\emptyset} \approx v_\emptyset$ and $v_{\theta,\text{cond}} \approx v_\emptyset + \alpha(v_{\text{cond}} - v_\emptyset)$, the effective field becomes:

$$v_{\text{eval}} = v_{\theta,\emptyset} + \beta\big(v_{\theta,\text{cond}} - v_{\theta,\emptyset}\big) \approx v_\emptyset + \alpha\beta\big(v_{\text{cond}} - v_\emptyset\big). \tag{44}$$

Thus, train-time guidance absorption and inference-time CFG scales compose multiplicatively. We include CFG absorption as an optional capability-field construction, not as the default objective for the main experiments.

7.8 Dense-Query Correlation

Suppose a dense trajectory objective queries K states from one student rollout and yields per-query gradient residuals $b_1, \ldots, b_K$. If the residuals were independent, averaging K queries would reduce variance by a factor of K. Along one rollout, however, residuals share the same noise seed, condition, student dynamics, and numerical path history. Let $\text{Var}(b_i) = \sigma_b^2$ and $\text{Corr}(b_i, b_j) = \rho$ for $i \neq j$. Then:

$$\text{Var}\left(\frac{1}{K}\sum_{i=1}^{K} b_i\right) = \frac{\sigma_b^2}{K}\big(1 + (K-1)\rho\big). \tag{45}$$

When $\rho \approx 0$, dense queries behave like independent supervision. When $\rho \approx 1$, the variance reduction nearly disappears. Worse, if several conflicting capability fields are accumulated in the same optimizer update, correlated queries can repeatedly amplify the same conflicting direction. This explains why dense querying is not automatically superior to a single semantic-side query. This connects the dense-query ablation to the third query choice in the main text. Increasing K changes the number of correlated states from the same rollout, whereas the default method sets K=1 to keep one teacher query per sample.

7.9 SDE Decorrelation

To test whether trajectory correlation is the failure mode, we inject stochasticity into the student rollout. Let σ_i be the scheduler noise scale at step i, let η control the injected stochasticity, and draw $\epsilon_i \sim \mathcal{N}(0, I)$:

$$z_{i+1} = z_i - \Delta t\, v_\theta(z_i, t_i, c) + \eta\, \sigma_i \sqrt{\Delta t}\, \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I). \tag{46}$$

This is an Euler Maruyama-style rollout, analogous to the stochastic processes used in score-based generative modeling [92]. Under a Lipschitz velocity field, independent noise injections perturb nearby rollout states by independent components. Consequently, residual correlations decay with temporal separation; informally, for constants $C_0, \kappa > 0$ depending on the local Lipschitz and noise scales:

$$|\text{Corr}(b_i, b_j)| \leq C_0 \exp\big(-\kappa\eta^2 |i - j|\big), \tag{47}$$

We use this only as a diagnostic mechanism: if SDE rollout rescues dense-query accumulation, it supports the hypothesis that correlated trajectory queries, rather than on-policy distillation itself, caused the degradation.

8 Implementations

This section covers additional details regarding implementations.

8.1 Computational Complexity

We report the wall-clock cost in Fig. 1 as seconds per training step under the same hardware setting. Table 4 decomposes the dominant per-step costs.

Table 4 Configuration of Per-Step Computational Complexity. All on-policy variants use the same N-step training rollout. The batch factor records extra micro-batches required by algorithmic grouping when the group does not fit in one physical micro-batch.

Method	Query State	Rollout	K	Batch Factor	Dominant Per-Step Cost
Off-Policy	offline noised state	none	K_{off}	γ_{std}	$K_{\text{off}} C_{\text{grad}}$
DiffusionOPD	student rollout	N-step ODE	K_{dense}	γ_{std}	$NC_{\text{roll}} + K_{\text{dense}} C_{\text{grad}}$
Flow-OPD	student rollout group	N-step SDE	K_{dense}	γ_{flow}	$\gamma_{\text{flow}}(NC_{\text{roll}} + K_{\text{dense}} C_{\text{grad}})$ plus PPO & log-prob overhead
DanceOPD	student rollout	N-step ODE	K_{ours}	γ_{std}	$NC_{\text{roll}} + K_{\text{ours}} C_{\text{grad}}$

The main algorithmic difference is whether a method first performs an on-policy student rollout, and how many rollout states contribute differentiable training signals. Let N denote the training rollout length, K the number of queried states used in the loss, B_{phys} the physical micro-batch size, G_{grp} the FlowGRPO group size, and γ the resulting micro-batch factor. We use C_{roll} for the cost of one no-gradient student rollout step and C_{grad} for one queried-state supervision unit, including the frozen teacher or reward query and the student forward and backward computation. In the timing setting of Fig. 1, N=16, B_{phys}=8, G_{grp}=16, K_{off}=K_{ours}=1, and K_{dense}=N; therefore γ_{std}=1 for methods without algorithmic grouping and γ_{flow}=$\lceil \frac{G_{\text{grp}}}{B_{\text{phys}}} \rceil$=2 for Flow-OPD. The rollout states are stop-gradient states; thus, K counts local gradient-bearing evaluations, not backpropagation through the numerical solver.

This decomposition explains the training-time pattern in Fig. 1. First, DanceOPD, DiffusionOPD, and Flow-OPD are all on-policy: with N=16 in our timing setting, they must roll out a complete N-step student trajectory before querying the teacher or reward signal. Off-policy distillation avoids this rollout and is therefore cheap per step, but it queries fixed noised endpoint states rather than the states visited by the current student.

Second, after the shared rollout, our model uses K_{ours}=1: it samples one semantic-side state and accumulates one local velocity-MSE gradient. DiffusionOPD instead supervises the full trajectory with K_{dense}=N=16, so the expensive teacher-query and backward part is multiplied by the rollout length.

Third, Flow-OPD uses the same dense K_{dense} trajectory signal but with FlowGRPO group size G_{grp}=16. Since the physical batch size is B_{phys}=8, one algorithmic group cannot fit in a single micro-batch and is split into γ_{flow}=2 micro-batches. This approximately doubles the wall-clock cost relative to the DiffusionOPD reproduction, with additional smaller overhead from SDE sampling, cached log-probabilities, and PPO clipping [57, 84]. Thus, the measured speed order in Fig. 1 follows the expected compute hierarchy: off-policy is cheapest but off-distribution, DanceOPD pays the rollout cost but only K_{ours} gradient state, DiffusionOPD pays dense K_{dense}-state gradients, and Flow-OPD further pays the FlowGRPO micro-batch factor.

8.2 DiffusionOPD

Because the DiffusionOPD [53] code was not open-sourced at the time of our experiments, we implemented a paper-faithful reproduction. The key reproduction details are as follows.

We reproduce the second-stage on-policy distillation recipe: task-specific teachers are treated as frozen capability fields, the current student first generates its own stop-gradient trajectory, and teacher supervision is evaluated on the student-visited states.

We use deterministic ODE rollout as the default DiffusionOPD sampler, following the broader fast-sampling view that diffusion and flow trajectories can be discretized by high-order solver choices [63, 109, 110]. Concretely, the student trajectory is generated by a 16-step Euler ODE rollout with no SDE noise, matching our 16-step training rollout. Unlike the main DanceOPD setting, which selects one semantic-side query state, the DiffusionOPD reproduction supervises the full rollout trajectory, i.e., K=16.

For a sample routed to teacher m, at each visited state $z_{t_j}^{\theta}$, we match the student and teacher transition

means. Let μ_θ and μ_m denote the corresponding student and teacher Euler transition means:

$$\mathcal{L}_{\text{DOPD-ODE}} = \sum_{j=1}^{K} \frac{1}{2} \left\| \mu_\theta(z_{t_j}^\theta) - \mu_m(z_{t_j}^\theta) \right\|_2^2, \tag{48}$$

where m is the routed capability teacher for the current sample. Under Euler flow matching, $\mu(z_t) = z_t - \Delta t\, v(z_t, t, c)$, so Eq. (48) is equivalent to:

$$\mathcal{L}_{\text{DOPD-ODE}} = \sum_{j=1}^{K} \frac{\Delta t^2}{2} \left\| v_\theta(z_{t_j}^\theta, t_j, c) - v_m(z_{t_j}^\theta, t_j, c) \right\|_2^2. \tag{49}$$

This keeps DiffusionOPD's ODE closed-form KL or transition-mean matching form rather than replacing it with our single-query MSE default.

We also reproduce DiffusionOPD's same-step multi-task accumulation at the capability-bucket level. The accumulated losses are averaged before the optimizer step so that the effective learning-rate scale is comparable across blocks.

8.3 Flow-OPD

We adapt the official Flow-OPD [24] code and recipe to our Z-Image setting. The reproduced core is on-policy SDE rollout with cached transition log-probabilities, routed teacher transition-KL rewards, and a PPO-style clipped objective [57, 84]. For each capability bucket, the student samples a global group of 16 SDE trajectories, uses K=16 queried states, SDE noise η=0.7, KL scale -1, and PPO clip range 10^{-4}. The Flow-OPD reproduction uses two active capability buckets and merge initialization for both composition blocks.

We disable MAR by setting β=0. Official MAR requires a separate frozen task-agnostic aesthetic teacher LoRA. We do not have a compatible Z-Image MAR teacher; the released Flow-OPD checkpoint is an SD3.5 final student adapter, and the style edit teacher is task-specific rather than a MAR anchor.

8.4 Off-Policy

The off-policy reproduction keeps the same hard-routed teacher and MSE loss as DanceOPD, but replaces the student-visited query state with a fixed noising distribution around offline endpoints, following the standard forward noising view of diffusion training [37]. For an edit sample, the endpoint is the encoded target image; for the prompt-only T2I bucket, no endpoint image is available, so the implementation uses a random latent endpoint. Let $\mathcal{E}$ denote the image encoder:

$$x_{0,m}^{\text{off}} = \begin{cases} \mathcal{E}(x_{\text{tgt}}), & m \in \{\text{Edit}, \text{Local}, \text{Global}\}, \\ \epsilon_0,\ \epsilon_0 \sim \mathcal{N}(0, I), & m = \text{T2I}. \end{cases} \tag{50}$$

We sample a fresh Gaussian noise vector and construct the grid query states using the same scheduler noising rule as SFT, where α_{t_j} and σ_{t_j} are scheduler coefficients:

$$z_{t_j}^{\text{off}} = \text{AddNoise}(x_{0,m}^{\text{off}}, \epsilon, t_j) \equiv \alpha_{t_j} x_{0,m}^{\text{off}} + \sigma_{t_j} \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{51}$$

Let c_m be the route-specific conditioning. The training objective is then plain velocity regression to the routed frozen teacher:

$$\mathcal{L}_{\text{off}} = \mathbb{E}_{m,(x,c),j,\epsilon,\epsilon_0} \left\| v_\theta(z_{t_j}^{\text{off}}, t_j, c_m) - \text{sg}\big[v_m(z_{t_j}^{\text{off}}, t_j, c_m)\big] \right\|_2^2. \tag{52}$$

Thus, this baseline changes only the query-state distribution: z_t^{off} comes from endpoint forward noising rather than the current student rollout z_t^θ. For this baseline, we use K=1, low-t Beta sampling, the same active route set and route probabilities as DanceOPD, and no KL, SDE, soft-teacher, or extrapolation terms.

Table 5 Default Hyper-Parameters. Unless otherwise stated, all Z-Image DanceOPD results use these settings.

Hyper-Parameter	Z-Image DanceOPD
Backbone	Z-Image [9]
Trainable module	DiT LoRA [38]
LoRA rank	128
Query state	student rollout state
Rollout steps	16-step Euler ODE
States per sample K	1
Route probabilities	uniform over active buckets
Timestep sampling	Beta$(5, 2)$ low-t bias
Objective	plain velocity MSE
Extrapolation scale	disabled
Optimizer	AdamW
Learning rate	2×10^{-4}
Gradient accumulation	4

8.5 On-Policy Generative Field Distillation

Z-Image DanceOPD Default Configuration. Z-Image [9] DanceOPD diagnostics in Fig. 8 panels (a) to (c), Fig. 9 panels (a) to (c), and Fig. 4 (b) use the same frozen capability fields unless otherwise stated: attribute and local edit, style and global edit, and the base T2I field. The student is initialized from the attribute-edit SFT LoRA and trained with hard-routed sample-wise velocity matching. Each optimizer step samples one capability route, rolls out the current student for 16 Euler steps, selects one semantic-side low-noise trajectory state, and minimizes plain velocity MSE to the routed frozen field. We disable extrapolation, delta clipping, EMA fake targets, DMD, SDS weighting, and consistency losses in the default setting; those variants are only used in ablations. Table 5 lists the default hyper-parameters.

Training Rollout. The 16-step rollout above is a stop-gradient query-state generator used during training, not a claim that the deployed sampler must use 16 steps. DanceOPD performs per-state velocity-field matching rather than trajectory-compression distillation, so the training query discretization and the benchmark inference discretization need not be identical [47, 62]. We use 16 steps for training because it gives on-policy coverage at a manageable cost; evaluations use the benchmark sampler stated in Sec. 8.6.

Tab. 2 Details For the T2I and Edit Composition block, there are two active buckets: one bucket is T2I, and the other bucket is Edit. The Edit bucket samples from both attribute and style edit data, but both are routed to the same joint Edit teacher; therefore, this setting is not a three-teacher update. For the Local and Global Edit Composition block, there are also two active buckets, with one Local bucket routed to the attribute teacher and one Global bucket routed to the style teacher. Unless otherwise stated, route probabilities are uniform over active buckets; the two main composition blocks therefore use a 1:1 route ratio, and the three-bucket diagnostics involving attribute, style, and T2I fields use a 1:1:1 route ratio.

Fig. 4 (c) Details The normalized trajectory coordinate is identical across the rollout-step variants: all rows sample $u \sim \mathrm{Beta}(5, 2)$, whose mass is biased toward the clean semantic end. The only change is the discretization grid size N: after sampling u, we quantize it by $\mathrm{idx} = \min(\lfloor uN \rfloor, N - 1)$. Thus, a larger N gives a finer low-t grid, but the same clean-side Beta mass is spread across more states, so the probability assigned to the single cleanest state decreases. For example, the cleanest state has probability $\Pr[\mathrm{idx} = N-1] = 1 - F_{\mathrm{Beta}(5,2)}(1 - \frac{1}{N})$. This is 0.167 for N=8 because idx 7 receives $u \in [\frac{7}{8}, 1]$, but only 0.017 for N=28 because idx 27 receives $u \in [\frac{27}{28}, 1]$; the remaining low-t mass is distributed over nearby clean-side states such as indices 20 to 27. Therefore, increasing the rollout length does not simply sample the cleanest state more often; it refines the clean-side coverage while reducing the mass on the terminal grid point.

Fig. 4 (a) SD3.5-M exhibits weak baseline realism and thus benefits from more pronounced improvements. The realism teacher is obtained by full-parameter training with SD3.5-M [21] for 100,000 steps, using a learning rate of 1×10^{-5} and a batch size of 16. Fig. 4 (a) then reports the subsequent OPD and distillation steps up

Table 6 CFG Absorption Diagnostics on GEditBench-EN. Training-time CFG scale α and inference-time CFG scale β compose approximately as $\alpha\beta$, but overly large training-time guidance can still damage the learned field.

Train α	Eval β	Eff. $\alpha\beta$	Subj-Add	Subj-Rep	Bg-Chg	Style-Chg	Color-Alt	Subj-Rem	Avg
3.5	1.0	3.5	6.485	5.897	4.824	5.907	5.628	3.790	5.422
1.0	7.0	7.0	6.266	6.181	5.924	5.060	5.716	5.357	5.751
3.5	2.0	7.0	6.553	6.507	5.911	5.814	5.998	4.215	5.833
2.0	7.0	14.0	4.684	5.402	4.114	3.750	4.313	5.115	4.563
7.0	7.0	49.0	3.645	4.367	4.090	3.321	3.902	4.765	4.015

to 3k, not the teacher-training step count. Our reward model is proprietary, which assigns a photorealism score to each input image, and we only use it for evaluation. For reward evaluation, we used 200 held-out samples excluded from the training data. Inference was performed at 1024×1024 resolution with 28 sampling steps and CFG scale 3.5.

Other Experiments Details For results in Tab. 8 and Tab. 9, we report checkpoints from 500 to 2000 steps. In Tab. 2, T2I refers to the Z-Image model. Local Edit is obtained by training OmniEdit [97] for one epoch on attribute subsets, Global Edit on style subsets, and Edit on the entire dataset.

8.6 Evaluation Settings

GEditBench-EN. For image-editing and edit-style capability composition, we follow the GEditBench-EN [60] protocol and report the six task categories used in the main table: subject addition, subject replacement, background change, style change, color alteration, and subject removal. The reported average is the arithmetic mean over these six categories, aligning with OmniEdit [97]'s six categories.

Broader image-generation evaluation has also studied reference-free caption alignment, question-answering faithfulness, human-preference rewards [49, 99, 100], compositional prompts, perceptual similarity, and multimodal or physical-world reasoning [15, 26, 34, 39, 40]. Unless otherwise specified, ablation scores are the GEditBench-EN average, because it is the most direct measurement of whether the distilled student preserves the routed editing capability. For the default non-CFG-sweep evaluations, we use 28 flow-matching sampling steps, and CFG scale 7.0.

GenEval. For testing T2I, we report the standard GenEval categories [29]: single object, two objects, counting, colors, position, and color attribution, together with the overall score. Rows whose target capability is not a T2I generation field may still include GenEval when preservation of the base model is part of the claim. For the default non-CFG-sweep evaluations, we use 28 flow-matching sampling steps, and CFG scale 3.5.

Table 7 Objective, Teacher-Schedule, and Dense-Query Diagnostics. All rows use the same capability fields and report GEditBench-EN sub-scores. The teacher-schedule column isolates how teacher signals enter an optimizer update: step alternation (G=1) routes one field per step, same-step accumulation (G=3) averages several routed losses before one update, and soft-teacher mixing is a non-routed objective diagnostic. The objective and query tag column records the loss family and, when applicable, dense-query aggregation and rollout control.

Teacher Schedule	Objective and Query Tag	Config	Subj-Add	Subj-Rep	Bg-Chg	Style-Chg	Color-Alt	Subj-Rem	Avg
Step alternation	MSE with ODE	K=1, G=1	6.266	6.181	5.924	5.060	5.716	5.357	5.751
Step alternation	Timestep-weighted MSE with ODE	K=1, G=1	5.953	6.151	5.400	3.986	5.334	6.729	5.592
Step alternation	DMD-EMA hybrid with ODE	K=1, G=1	5.731	6.219	5.093	4.752	5.558	6.228	5.597
Step alternation	SDS+DMD-EMA with ODE	K=1, G=1	6.397	6.032	5.037	4.040	5.289	5.943	5.456
Step alternation	Consistency with ODE	K=1, G=1	5.929	6.039	5.117	5.306	4.907	5.843	5.523
Step alternation	KL-$\bar{\sigma}^2$ with ODE	K=1, G=1	5.906	6.061	5.436	4.910	5.427	5.268	5.501
Step alternation	DMD2-inter with ODE	K=1, G=1	5.526	5.597	4.975	4.883	5.231	3.637	4.975
Step alternation	DMD2-dist with ODE	K=1, G=1	5.596	5.856	4.345	3.016	5.060	5.124	4.833
Step alternation	AuxFeat with ODE	K=1, G=1	4.864	5.568	4.850	3.885	4.019	5.303	4.748
Soft-teacher mixing	Soft-teacher MSE with ODE	K=1, all teachers	5.410	5.808	4.902	4.426	5.191	4.225	4.994
Soft-teacher mixing	Soft-teacher KL-$\bar{\sigma}^2$ with ODE	K=1, all teachers	5.302	5.626	4.991	4.290	5.042	4.604	4.976
Same-step accumulation	MSE with ODE	K=1, G=3	5.170	5.754	5.754	5.427	6.171	4.632	5.485
Same-step accumulation	MSE with SDE	K=1, G=3, η=0.3	4.862	5.068	4.358	4.719	4.816	4.048	4.645
Step alternation	MSE with ODE, sum	K=2, G=1	6.339	6.075	5.573	3.159	5.792	5.955	5.482
Step alternation	MSE with SDE, sum	K=2, G=1, η=0.3	5.366	5.856	4.461	3.882	5.205	5.381	5.025
Step alternation	MSE with ODE, weighted	K=2, G=1	4.850	5.873	5.132	4.094	4.522	5.118	4.931
Step alternation	MSE with ODE, weighted	K=4, G=1	5.962	5.886	5.500	5.065	5.654	3.912	5.330
Step alternation	MSE with ODE, isolation	K=4, G=1	5.476	5.654	4.995	4.540	5.138	4.066	4.978
Step alternation	MSE with ODE, isolation	K=8, G=1	6.207	5.473	4.687	3.716	4.953	5.837	5.145
Step alternation	MSE with ODE, weighted	K=8, G=1	5.224	5.877	5.572	4.085	4.821	5.728	5.218
Step alternation	MSE with ODE, sum no control	K=8, G=1	5.913	5.744	4.620	3.723	4.817	5.409	5.038
Step alternation	MSE with ODE, weighted	K=16, G=1	5.638	6.062	4.776	3.878	5.181	5.227	5.127
Same-step accumulation	MSE with ODE	K=2, G=3	4.458	5.193	5.265	4.144	4.672	2.894	4.437
Same-step accumulation	MSE with SDE	K=2, G=3, η=0.3	5.751	5.994	5.832	4.114	5.153	4.689	5.255
Same-step accumulation	MSE with SDE	K=16, G=3, η=0.3	5.093	5.422	4.583	3.029	4.685	5.939	4.792

Table 8 Timestep and Initialization Ablations. Scores are GEditBench-EN sub-scores unless otherwise noted. Objective, teacher-schedule, dense-query, and rollout-control diagnostics are consolidated in Table 7. These ablations support semantic-side low-t querying and the default initialization choice.

Axis	Variant	Step	Subj-Add	Subj-Rep	Bg-Chg	Style-Chg	Color-Alt	Subj-Rem	Avg
Timestep query	Low-t	500	5.776	5.759	5.371	4.148	5.033	4.728	5.136
Timestep query	Low-t	1000	5.543	5.725	5.327	4.515	5.445	5.586	5.357
Timestep query	Low-t	1500	5.233	5.787	5.198	4.394	5.208	5.161	5.163
Timestep query	Low-t	2000	6.266	6.181	5.924	5.060	5.716	5.357	5.751
Timestep query	Median-t	500	5.626	5.902	4.862	4.387	4.991	5.781	5.258
Timestep query	Median-t	1000	5.548	5.779	4.951	4.538	5.383	4.393	5.099
Timestep query	Median-t	1500	5.405	5.567	4.576	4.914	5.485	4.233	5.030
Timestep query	Median-t	2000	4.611	5.656	4.353	4.496	5.012	3.764	4.649
Timestep query	High-t	500	5.698	5.959	5.113	5.359	5.292	5.099	5.420
Timestep query	High-t	1000	4.524	4.580	4.801	2.642	3.978	3.727	4.042
Timestep query	High-t	1500	3.473	4.030	3.947	3.129	4.287	3.480	3.724
Timestep query	High-t	2000	4.290	5.485	5.186	3.221	5.166	5.528	4.813
Initialization	Merged	500	3.441	4.395	4.752	5.445	5.240	2.002	4.213
Initialization	Merged	1000	3.113	4.269	3.119	4.008	3.821	1.040	3.228
Initialization	Merged	1500	4.485	5.279	5.091	5.023	4.999	1.991	4.478
Initialization	Merged	2000	4.212	4.680	4.603	4.845	4.426	2.392	4.193
Initialization	T2I	500	1.443	1.933	2.384	2.088	1.316	0.157	1.554
Initialization	T2I	1000	1.912	2.980	2.869	2.301	2.410	0.166	2.106
Initialization	T2I	1500	2.006	3.205	2.856	2.290	2.272	0.336	2.161
Initialization	T2I	2000	1.571	2.338	3.289	2.024	1.837	0.276	1.889
Initialization	Local Edit	500	5.776	5.759	5.371	4.148	5.033	4.728	5.136
Initialization	Local Edit	1000	5.543	5.725	5.327	4.515	5.445	5.586	5.357
Initialization	Local Edit	1500	5.233	5.787	5.198	4.394	5.208	5.161	5.163
Initialization	Local Edit	2000	6.266	6.181	5.924	5.060	5.716	5.357	5.751
Initialization	Global Edit	500	3.459	4.670	4.903	5.406	4.348	1.675	4.077
Initialization	Global Edit	1000	2.922	4.178	4.644	4.563	4.247	0.702	3.543
Initialization	Global Edit	1500	3.563	4.349	5.132	5.182	3.958	1.246	3.905
Initialization	Global Edit	2000	2.609	3.887	3.661	3.112	2.643	0.300	2.702

Table 9 Training Rollout-Step Sensitivity. All rows use hard-routed MSE with ODE with K=1, G=1, low-t Beta$(5, 2)$ query sampling, and the same evaluation protocol. Only the number of stop-gradient student-rollout steps used to generate training query states is varied. We report both GEditBench-EN and GenEval because rollout discretization may affect edit quality and base T2I preservation differently.

		GEditBench-EN							GenEval
Rollout	Step	Subj-Add	Subj-Rep	Bg-Chg	Style-Chg	Color-Alt	Subj-Rem	Avg	Overall
8	500	5.067	5.237	4.372	4.184	4.564	3.611	4.506	0.833
8	1000	5.237	5.819	4.735	4.500	4.962	4.271	4.921	0.855
8	1500	5.500	5.788	5.559	4.846	5.079	4.642	5.236	0.866
8	2000	6.514	6.137	5.532	5.163	6.346	4.744	5.739	0.852
16	500	5.776	5.759	5.371	4.148	5.033	4.728	5.136	0.821
16	1000	5.543	5.725	5.327	4.515	5.445	5.586	5.357	0.862
16	1500	5.233	5.787	5.198	4.394	5.208	5.161	5.163	0.854
16	2000	6.266	6.181	5.924	5.060	5.716	5.357	5.751	0.858
20	500	6.045	5.678	5.089	4.180	4.838	6.144	5.329	0.832
20	1000	5.995	5.660	6.109	4.733	5.833	5.570	5.650	0.855
20	1500	5.339	5.603	4.958	5.034	5.237	3.916	5.014	0.846
20	2000	5.889	5.899	5.440	5.042	5.696	5.531	5.583	0.842
28	500	4.208	5.271	4.338	3.686	4.323	2.779	4.101	0.849
28	1000	3.983	5.175	3.996	3.120	4.228	2.527	3.838	0.866
28	1500	4.523	5.409	4.699	4.220	4.697	3.157	4.451	0.851
28	2000	6.544	6.147	5.618	5.393	6.475	4.008	5.697	0.834

9 Additional Qualitative Results

Global Scene and Style Edits. Fig. 12 compares large transformation edits where both the global appearance and the original scene identity must be controlled.

In the Venetian canal example, the requested change is from golden-hour dusk to a silent snowy night. DanceOPD converts the warm canal scene into a coherent blue night scene with visible snow, snow-covered walkways, and lamp illumination, while preserving the canal geometry and the gondola-like foreground structure. Several baselines either under-transform the scene, introduce excessive blur, or lose part of the original spatial structure.

In the portrait example, the target cyberpunk style requires strong neon lighting, a darker futuristic environment, and visible local accessories. Our model applies the magenta and cyan cyberpunk appearance while retaining the main face, pose, hair layout, and portrait composition more consistently than the alternatives.

Local Attribute Changes and Re-Staging. Fig. 13 shows two cases that require different preservation and transformation balances.

For the evening-gown edit, the main object and the indoor lighting should remain stable, while the dress changes from crimson silk to emerald-green velvet. DanceOPD changes both the color and material impression of the gown, preserving the human pose and the hall background; in contrast, some baselines either keep a weaker material change, flatten the green dress texture, or alter the image style more globally.

For the rental-room edit, the target is broader: the room should become tidier and re-staged while still looking like the same indoor space. Our model removes much of the clutter and produces a cleaner room layout without turning the example into an unrelated scene, whereas other methods show stronger layout drift, collage-like structure changes, or incomplete tidying.

Multiple Edits on a Shared Source Object. Fig. 14 edits the same water-bottle source with six different instructions. The results cover local surface changes, such as condensation drops and rose-gold metallic finish; decorative restyling, such as the cherry-blossom edition; context changes, such as placing the bottle in a mountain hiking scene; structural reinterpretation, such as the technical exploded diagram; and material transformation, such as a transparent glass reveal.

Across these edits, DanceOPD generally preserves the bottle identity, cap shape, upright product composition,

and central placement while changing the requested visual factor, a regime closely related to object binding, structured guidance, and composable-condition control in text-to-image generation [10, 25, 41]. This illustrates that the distilled student is not merely memorizing one edit mode: it can reuse the shared object representation across diverse routed capability queries.

10 Additional Quantitative Results

Tables 6 to 9 provide the detailed numeric results behind the main CFG, routing, objective, timestep, initialization, and rollout-step analyses.

Diagnostic Scope. These tables should be read as controlled diagnostics rather than as additional main-result blocks. Table 2 compares complete composition settings, whereas Tables 6 to 9 isolate individual implementation and estimator choices under matched capability fields.

Consequently, the same default diagnostic anchor can appear in multiple places: hard-routed MSE with ODE, K=1, G=1, a 16-step training rollout, and low-t query sampling. Repeated rows or repeated averages should therefore be interpreted as shared controls for ablations, not as independent duplicates of the Table 2 main settings.

CFG Absorption Diagnostics. Table 6 separates training-time absorbed guidance from inference-time external CFG. The column α denotes the guidance scale used to define the target field during absorption, while β denotes the guidance scale applied again at evaluation.

Under the affine CFG approximation in Sec. 7.7, the effective guidance strength is roughly $\alpha\beta$. Thus, α=1, β=7 is the eval-only CFG control, α=3.5, β=1 measures train-only absorption, and α=3.5, β=2 tests a moderate composition of absorbed and external guidance. The large-drop rows illustrate the expected over-guidance failure mode when the effective scale becomes too high.

Rollout-Step Diagnostics. Table 9 varies only the number of stop-gradient student-rollout steps used to generate training query states.

The benchmark evaluation sampler is held fixed, so the rows should not be read as evaluating the final model with different sampling-step budgets. Because all rows use the same Beta(5, 2) distribution over the normalized semantic coordinate, increasing the rollout length refines the clean-side grid but also spreads the same probability mass over more candidate states.

This is why longer training rollouts do not automatically improve GEditBench or GenEval: the rollout is a query-state generator for field matching, not a trajectory-compression target.

Figure 12 T2I and Edit Composition: Global Edits. DanceOPD better follows large style and scene transformations while preserving image structure compared with off-policy distillation, joint training, DiffusionOPD, and Flow-OPD.

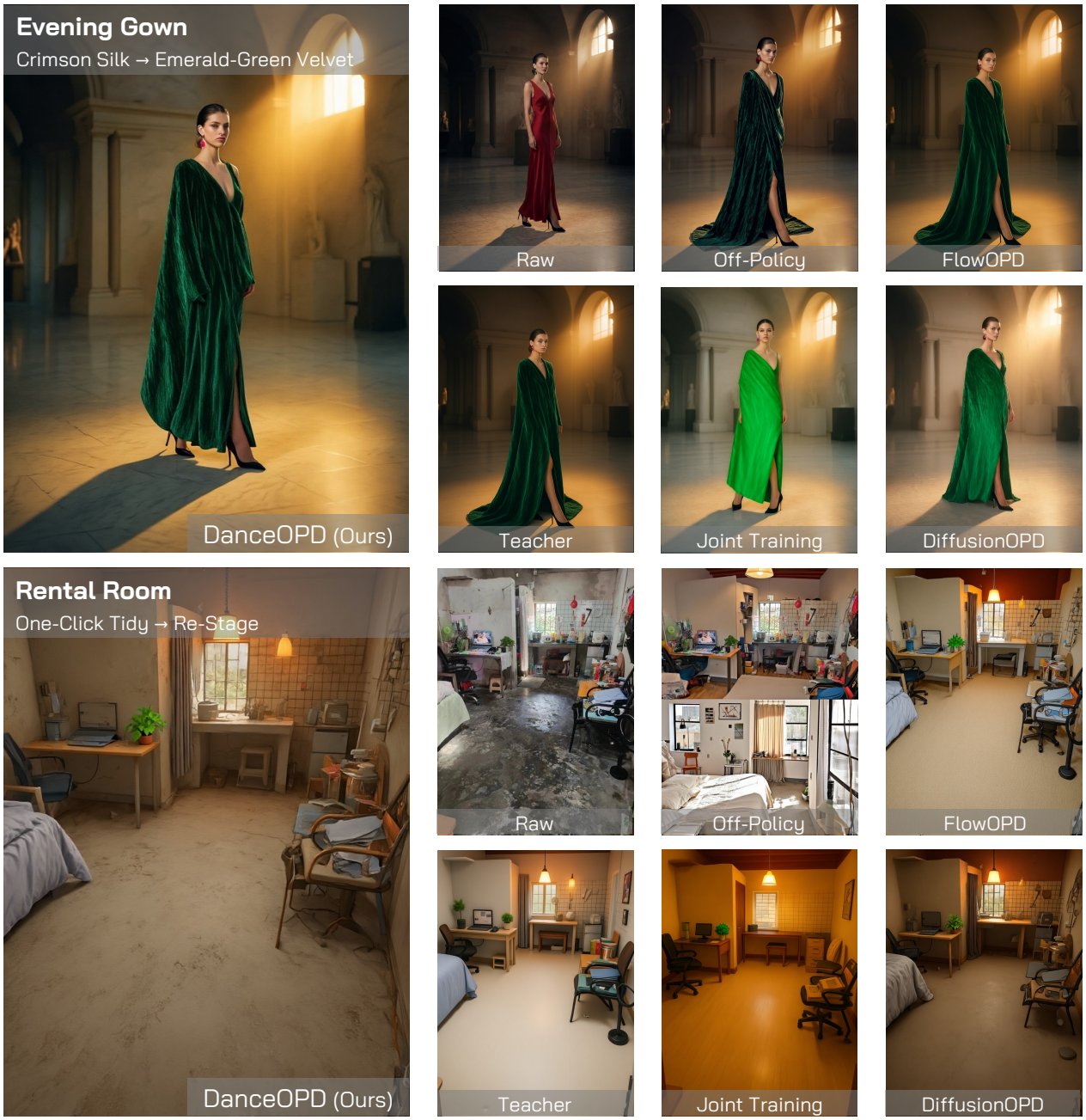

Figure 13 T2I and Edit Composition: Local and Global Edits. DanceOPD preserves content under local edits and produces stronger global transformations than the competing baselines.

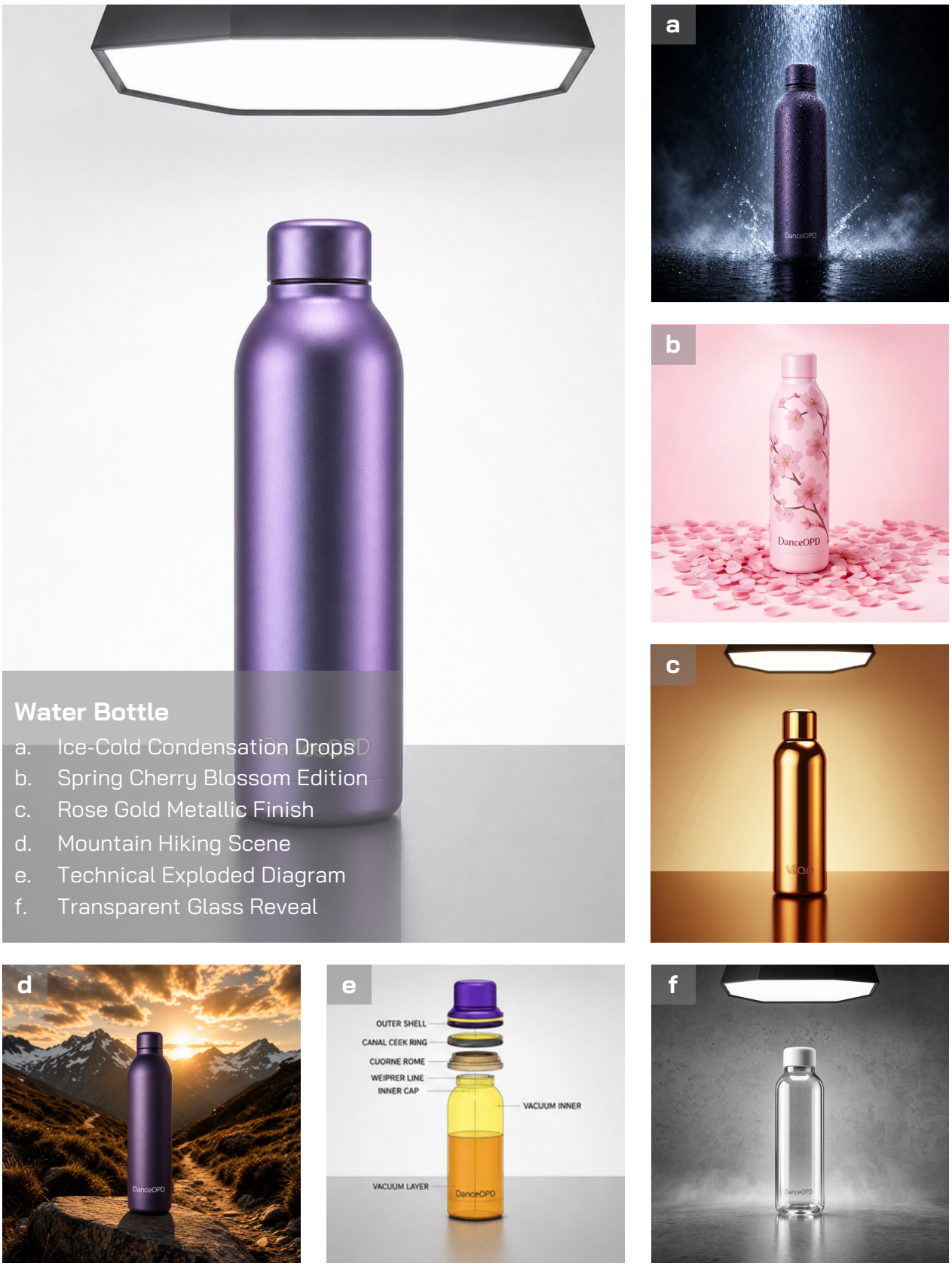

Figure 14 T2I and Edit Composition: Various Edits on the Same Object. DanceOPD supports diverse object-level transformations on the same source object, including material, style, scene, structure, and transparency edits.

References

[1] 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, pages 21246–21263, 2024.

[2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9163–9171, 2019.

[3] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. arXiv preprint arXiv:2209.04836, 2022.

[4] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. Journal of Machine Learning Research, 26(209):1–80, 2025.

[5] 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.

[6] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In International Conference on Machine Learning, pages 1737–1752. PMLR, 2023.

[7] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In International Conference on Learning Representations, volume 2024, 2024.

[8] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18392–18402, 2023.

[9] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699, 2025.

[10] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM Transactions on Graphics, 42(4):1–10, 2023.

[11] Sherry X Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. TINO-Edit: Timestep and noise optimization for robust diffusion-based image editing. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6337–6346, 2024.

[12] Sixiang Chen, Jinbin Bai, Zhuoran Zhao, Tian Ye, Qingyu Shi, Donghao Zhou, Wenhao Chai, Xin Lin, Jianzong Wu, Chao Tang, et al. An empirical study of GPT-4o image generation capabilities. arXiv preprint arXiv:2504.05979, 2025.

[13] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In International Conference on Machine Learning, pages 794–803. PMLR, 2018.

[14] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. Advances in Neural Information Processing Systems, 33:2039–2050, 2020.

[15] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Campagnolo Guizilini, and Yue Wang. PhysBench: Benchmarking and enhancing vision-language models for physical world understanding. In International Conference on Learning Representations, 2025.

[16] Wei Chow, Linfeng Li, Lingdong Kong, Zefeng Li, Qi Xu, Hang Song, Tian Ye, Xian Wang, Jinbin Bai, Shilin Xu, Xiangtai Li, Junting Pan, Shaoteng Liu, Ran Zhou, Tianshu Yang, and Songhua Liu. EditMGT: Unleashing potentials of masked generative transformers in image editing. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 38038–38048, 2026.

[17] Wei Chow, Linfeng Li, Xian Sun, Lingdong Kong, Zefeng Li, Qi Xu, Hang Song, Tian Ye, Xian Wang, Jinbin Bai, Shilin Xu, Xiangtai Li, Junting Pan, Shaoteng Liu, Ran Zhou, Tianshu Yang, and Songhua Liu. Masked generative transformer is what you need for image editing. arXiv preprint arXiv:2605.10859, 2026.

[18] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. arXiv preprint arXiv:2307.08698, 2023.

[19] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In Advances in Neural Information Processing Systems, volume 34, pages 8780–8794, 2021.

[20] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In International Conference on Machine Learning, pages 8489–8510. PMLR, 2023.

[21] 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 International Conference on Machine Learning, pages 12606–12633. PMLR, 2024.

[22] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. Advances in Neural Information Processing Systems, 36:79858–79885, 2023.

[23] Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation. arXiv preprint arXiv:2605.07396, 2026.

[24] 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.

[25] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. arXiv preprint arXiv:2212.05032, 2022.

[26] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. arXiv preprint arXiv:2306.09344, 2023.

[27] Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In Interspeech, pages 3697–3701, 2017.

[28] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618, 2022.

[29] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. Advances in Neural Information Processing Systems, 36:52132–52152, 2023.

[30] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In International Conference on Learning Representations, volume 2024, pages 32694–32717, 2024.

[31] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-SNR weighting strategy. In IEEE/CVF International Conference on Computer Vision, pages 7441–7451, 2023.

[32] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In IEEE/CVF International Conference on Computer Vision, pages 1921–1930, 2019.

[33] 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.

[34] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Conference on Empirical Methods in Natural Language Processing, pages 7514–7528, 2021.

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.

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

[38] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022.

[39] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In IEEE/CVF International Conference on Computer Vision, pages 20406–20417, 2023.

[40] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 47(5):3563–3579, 2025.

[41] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. arXiv preprint arXiv:2302.09778, 2023.

[42] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. arXiv preprint arXiv:2212.04089, 2022.

[43] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. Neural computation, 3(1):79–87, 1991.

[44] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. arXiv preprint arXiv:2103.02631, 2021.

[45] Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. arXiv preprint arXiv:2605.06387, 2026.

[46] 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.

[47] 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.

[48] Dongjun Kim, Chieh-Hsin Lai, WeiHsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In International Conference on Learning Representations, 2024.

[49] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-A-Pic: An open dataset of user preferences for text-to-image generation. In Advances in Neural Information Processing Systems, volume 36, pages 36652–36663, 2023.

[50] Lingdong Kong, Xian Sun, Wei Chow, Linfeng Li, Kevin Qinghong Lin, Xuan Billy Zhang, Song Wang, Rong Li, Qing Wu, Wei Gao, et al. AI for auto-research: Roadmap & user guide. arXiv preprint arXiv:2605.18661, 2026.

[51] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VieScore: Towards explainable metrics for conditional image synthesis evaluation. In Annual Meeting of the Association for Computational Linguistics, pages 12268–12290, 2024.

[52] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1931–1941, 2023.

[53] 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.

[54] Haonan Lin, Yan Chen, Jiahao Wang, Wenbin An, Mengmeng Wang, Feng Tian, Yong Liu, Guang Dai, Jingdong Wang, and Qianying Wang. Schedule your edit: A simple yet effective diffusion noise schedule for image editing. Advances in Neural Information Processing Systems, 37:115712–115756, 2024.

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

[56] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems, 34:18878–18890, 2021.

[57] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. Advances in Neural Information Processing Systems, 38:40783–40818, 2026.

[58] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In International Conference on Learning Representations, 2021.

[59] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In European Conference on Computer Vision, pages 423–439. Springer, 2022.

[60] 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.

[61] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003, 2022.

[62] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems, 35:5775–5787, 2022.

[63] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. Machine Intelligence Research, 22(4):730–751, 2025.

[64] Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying OPD: Length inflation and stabilization strategies for large language models. arXiv preprint arXiv:2604.08527, 2026.

[65] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378, 2023.

[66] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1930–1939, 2018.

[67] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In Annual Meeting of the Association for Computational Linguistics, pages 565–576, 2021.

[68] Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. Advances in Neural Information Processing Systems, 35:17703–17716, 2022.

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

[70] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. arXiv preprint arXiv:2202.01017, 2022.

[71] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning, pages 8162–8171. PMLR, 2021.

[72] William Peebles and Saining Xie. Scalable diffusion models with transformers. In IEEE/CVF International Conference on Computer Vision, pages 4195–4205, 2023.

[73] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In Conference of the European Chapter of the Association for Computational Linguistics, pages 487–503, 2021.

[74] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In International Conference on Learning Representations, 2024.

[75] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3D using 2D diffusion. arXiv preprint arXiv:2209.14988, 2022.

[76] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. arXiv preprint arXiv:2310.03739, 2023.

[77] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10684–10695, 2022.

[78] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[79] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In International Conference on Artificial Intelligence and Statistics, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

[81] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512, 2022.

[82] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In SIGGRAPH Asia, pages 1–11, 2024.

[83] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In European Conference on Computer Vision, pages 87–103. Springer, 2024.

[84] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[85] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Advances in Neural Information Processing Systems, volume 31, 2018.

[86] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. ZipLoRA: Any subject in any style by effectively merging LoRAs. In European Conference on Computer Vision, pages 422–438. Springer, 2024.

[87] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538, 2017.

[88] Chengchao Shen, Mengqi Xue, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In IEEE/CVF International Conference on Computer Vision, pages 3504–3513, 2019.

[89] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pages 2256–2265. PMLR, 2015.

[90] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

[91] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Advances in Neural Information Processing Systems, volume 32, 2019.

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

[93] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In International Conference on Machine Learning, pages 32211–32252. PMLR, 2023.

[94] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. arXiv preprint arXiv:2302.00482, 2023.

[95] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8228–8238, 2024.

[96] Luozhou Wang, Shuai Yang, Shu Liu, and Ying-cong Chen. Not all steps are created equal: Selective diffusion distillation for image manipulation. In IEEE/CVF International Conference on Computer Vision, pages 7472–7481, 2023.

[97] 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, pages 259–271, 2025.

[98] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In International Conference on Machine Learning, pages 23965–23998. PMLR, 2022.

[99] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In IEEE/CVF International Conference on Computer Vision, pages 2096–2105, 2023.

[100] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In Advances in Neural Information Processing Systems, volume 36, pages 15903–15935, 2023.

[101] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. Advances in Neural Information Processing Systems, 36:7093–7115, 2023.

[102] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In International Conference on Learning Representations, volume 2024, pages 22743–22763, 2024.

[103] Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. arXiv preprint arXiv:2602.12125, 2026.

[104] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6613–6623, 2024.

[105] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1285–1294, 2017.

[106] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. AnyEdit: Mastering unified high-quality image editing for any idea. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 26125–26135, 2025.

[107] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. Advances in Neural Information Processing Systems, 33:5824–5836, 2020.

[108] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928, 2016.

[109] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. arXiv preprint arXiv:2204.13902, 2022.

[110] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. Advances in Neural Information Processing Systems, 36: 49842–49869, 2023.

[111] Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. arXiv preprint arXiv:2605.27028, 2026.