CAPFIELD-OPD: LEARNING CONTINUOUS CAPABILITY FIELDS VIA JOINT-ANCHORED MULTI-TEACHER ON-POLICY DISTILLATION FOR FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-specialized post-training produces strong experts for flow-based generative models, while multi-teacher on-policy distillation (OPD) consolidates their capabilities into a single student. Existing methods, however, route each prompt to a single teacher according to its semantic category, implicitly binding the desired capability to prompt content. This coupling makes capability invocation vulnerable to prompt perturbations and prevents users from explicitly adjusting the strength of the desired capability at inference time. In this work, we introduce CapField-OPD, an OPD framework that integrates multiple teachers into a continuous capability field through explicit capability coordinates. We use teacher models as anchors to construct this field, with the coordinates determining how their outputs are combined. Each capability configuration thus receives a unique supervision target, and capability control no longer depends on prompt semantics. Since the training anchors may not be optimal at inference time, we further profile the learned field on a small calibration set. The coordinate with the highest mean reward serves as the recommended default, while coordinates that are frequently optimal offer a promising candidate set for test-time scaling. Extensive experiments on compositional generation, text rendering, and visual aesthetics demonstrate that CapField-OPD consolidates multiple specialized teachers into a single student while preserving or surpassing their performance, reliably invokes the desired capabilities under semantics-preserving prompt variations, and supports continuous capability control and coordinate-based test-time scaling.

1 INTRODUCTION

Flow-matching models (Esser et al., 2024; Lipman et al., 2022; Liu et al., 2022) have recently become a popular framework for image generation (Labs, 2024; Team, 2025; Wu et al., 2025), in which a learned velocity field can transport Gaussian noise to data samples via iterative denoising. Reward-based post-training (Liu et al., 2025a; Xue et al., 2025b; Zhou et al., 2026b) further improves specific generation capabilities, such as text-rendering accuracy, compositional fidelity, and visual aesthetics, producing strong and diverse domain experts. However, training and deploying a separate model for each capability is costly. Multi-teacher on-policy distillation (OPD) (Fang et al., 2026) addresses this problem by consolidating reward-specialized teachers into a single student, which is supervised by teachers' outputs evaluated along student-generated trajectories.

Nevertheless, existing flow-based multi-teacher OPD methods (Fang et al., 2026; Li et al., 2026c; Zhou et al., 2026a) typically split training prompts by task type and assign each subset to one teacher via semantics-driven hard routing. Without an explicit capability signal, the student must infer the desired capability category from prompt semantics, binding capability intent to prompt content. Capability activation thus becomes sensitive to prompt wording and style, especially on out-of-distribution prompts, and users cannot explicitly control capabilities at inference time. As shown in Fig. 1(b), for the same prompt describing a handwritten sign reading "JAZZ LEGENDS ON VINYL HERE," one user may prioritize accurate text rendering, another may prefer visual aesthetics, while a third may require both with different strengths. A semantic router cannot distinguish these intentions from the prompt alone. When a prompt requires multiple capabilities, naively activating multiple

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

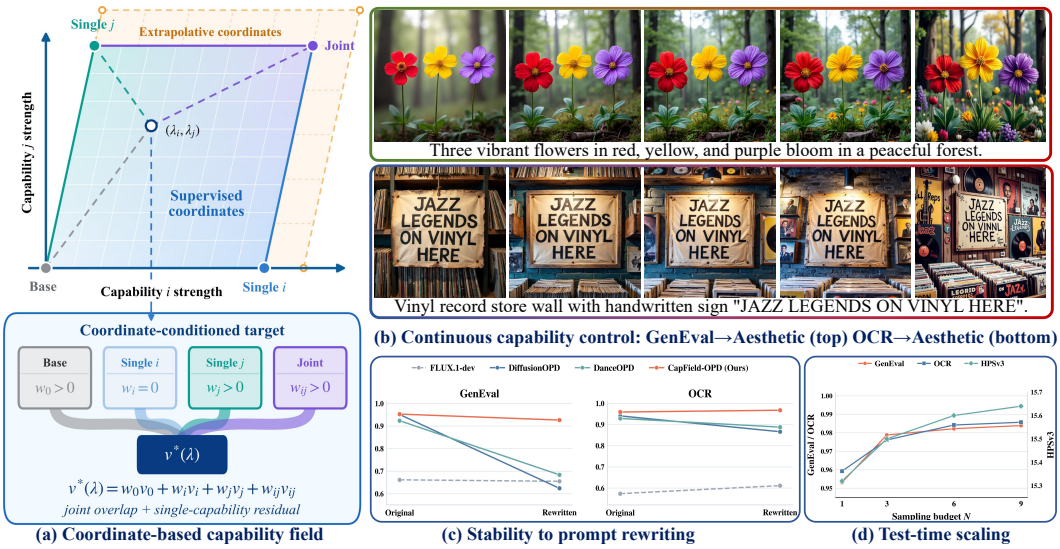

Figure 1: Overview of CapField-OPD. (a) Teacher anchors define a continuous capability field controlled by explicit, potentially extrapolative coordinates. (b-d) The learned field supports continuous control, robust capability invocation under prompt rewriting, and coordinate-based test-time scaling.

teachers yields conflicting velocity targets under the same student condition. These limitations point to a key issue: capability intent requires explicit representation and control.

To this end, we propose CapField-OPD, built on the principle that desired capabilities should be explicitly specified rather than implicitly inferred. Specifically, we condition the student on an explicit capability coordinate whose axes control capability strengths, and reinterpret the base, single-capability, and joint-capability teachers as anchors of a shared capability field. A joint-anchored target field combines their outputs through coordinate-dependent activation weights. For coordinates activating multiple capabilities, the corresponding joint-capability teacher supplies the shared capability components, and remaining activation is allocated to the single-capability teachers, so the field exactly preserves each anchor's capability while providing a continuous velocity target over the capability space. The student can thus invoke desired capabilities even on boundary or out-of-distribution prompts where semantic routing may fail (Fig. 1(c)). Coordinate conditioning also adds an inference-time scaling axis. Although supervision spans a bounded range, varying the coordinates teaches the student how its velocity field changes as capabilities strengthen or combine. The teacher anchors therefore define reference states rather than hard performance limits. As shown in Fig. 1(a), extending coordinates beyond the training range continues the learned capability response, and this bounded extrapolation can yield higher rewards than the teacher anchors. To identify reliable extrapolative settings, we profile in-range and extrapolative coordinates on a small calibration set of prompt–noise pairs. The coordinate with the highest mean reward serves as the default, and frequently optimal coordinates provide candidates for prompt-specific search, enabling coordinate-based test-time scaling (Fig. 1(d)).

Our contributions are as follows: (1) We identify semantic-capability entanglement of hard routing as a central limitation of multi-teacher OPD. (2) We propose CapField-OPD, which learns a continuous capability field through explicit capability conditioning and joint-anchored field supervision, with capability landscape profiling for default-coordinate selection and test-time search. (3) Experiments show CapField-OPD consolidates reward-specialized experts, enhances robustness to prompt perturbations, and supports continuous capability control and coordinate-based test-time scaling.

2 Related Work

2.1 Preference Alignment in Flow Models

Aligning pretrained flow models with human preferences (Liu et al., 2025b; Xue et al., 2025b; Bu et al., 2026) has emerged as an effective post-training strategy for adapting them to downstream

objectives. Existing approaches can be broadly divided into offline preference learning and online reward optimization. Offline DPO-style methods (Wallace et al., 2024; Yang et al., 2024) translate preferred–dispreferred comparisons into flow-matching objectives (Liu et al., 2025b), thereby avoiding explicit reward-model training and online rollouts. Online methods (Liu et al., 2025a; Ling et al., 2026; Li et al., 2026b), in contrast, formulate the sampling trajectory as a Markov decision process (MDP) and interleave trajectory sampling and reward scoring with policy updates. Specifically, the PPO-style methods (DPOK (Fan et al., 2023), DDPO (Black et al., 2023)) apply policy gradients along denoising trajectories, whereas GRPO-style approaches (Liu et al., 2025a; Li et al., 2025; Wang & Yu, 2025) convert an ordinary differential equation (ODE) sampling into an equivalent stochastic differential equation (SDE) and optimize the sampling distribution using estimated relative advantages. Both of them rely on tractable stochastic transition densities and perform on-policy optimization along the sampled trajectories. In comparison, DiffusionNFT (Zheng et al., 2026) and Advantage Weighted Matching (AWM) (Xue et al., 2025a) develop a different approach. They collect and score generated images, and then re-noise them for reward-derived weighted optimization, achieving substantially faster convergence. Collectively, these techniques improve the performance of preference alignment toward specific reward functions.

2.2 On-policy distillation in flow models

Unlike conventional distillation methods (Yin et al., 2024a;b; Cheng et al., 2025; Chen et al., 2025) that primarily aim to reduce sampling steps, on-policy distillation (OPD) (Fang et al., 2026; Li et al., 2026c; Zhou et al., 2026a) has recently emerged as an effective approach for consolidating the capabilities of multiple teachers into a single student model. Instead of relying on fixed data or teacher-generated results, OPD samples trajectories from the current student and queries teachers at student-visited states, reducing exposure bias while providing dense velocity-field supervision. Specifically, Flow-OPD (Fang et al., 2026) and DiffusionOPD (Li et al., 2026c) independently train reward-specialized teachers and consolidate their capabilities into a unified student through task routing and trajectory-level matching. Subsequent works extend OPD in several directions. For example, DreOPD (Lin et al., 2026) moves beyond direct teacher imitation through degraded-reference velocity extrapolation, while Any-OPD (Fu et al., 2026) enables distillation between heterogeneous teacher–student pairs through representation-space alignment. In addition, CFG-OPD (Li et al., 2026a) separately constrains the positive prediction and the CFG direction, reducing sensitivity to guidance scales. Although these methods improve OPD for flow models, existing multi-teacher OPD methods still rely on semantics-driven hard routing, assigning each trajectory to one expert. This entangles prompt content with capability intent, limiting capability composition and strength control and causing conflicting supervision for multi-capability or boundary prompts.

3 Method

3.1 Preliminaries

Flow Matching Models. Flow matching (Lipman et al., 2022; Liu et al., 2022) learns a time-dependent velocity field that transports Gaussian noise to clean data. Given a paired sample $(\mathbf{x}, c) \sim p_{\text{data}}$, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and a timestep $t \sim \mathcal{U}[0, 1]$, the interpolated noisy state is $\mathbf{x}_t = (1-t)\mathbf{x} + t\boldsymbol{\epsilon}$, where $\mathbf{x}_0 = \mathbf{x}$ is the clean sample and $\mathbf{x}_1 = \boldsymbol{\epsilon}$ is pure noise. Differentiating the path with respect to t gives the target velocity

$$v_t = \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \boldsymbol{\epsilon} - \mathbf{x}. \tag{1}$$

The flow model $v_\theta(\mathbf{x}_t, t, c)$ is then trained with the flow matching loss

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{(\mathbf{x},c),\boldsymbol{\epsilon},t}\left[\|v_\theta(\mathbf{x}_t, t, c) - v_t\|_2^2\right], \tag{2}$$

where c is an optional condition such as text or image.

On-Policy Distillation. On-policy distillation (OPD) (Li et al., 2026c) aims to transfer the capabilities of frozen teachers to a student by imitating the teacher's behavior on student-visited states, thereby offering supervision that adapts to the student's evolving distribution. Let v_ϕ and v_θ denote

the teacher and student velocity fields, respectively, and let $\tau = \{\mathbf{x}_{t_i}\}_{j=0}^{M}$ be the trajectory sampled by the student model; the OPD objective is defined as:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{c,\tau\sim p_\theta(\tau|c)}\left[\sum_{i=0}^{M-1} w(t_i)\,\|v_\theta(\mathbf{x}_{t_i}, t_i, c) - v_\phi(\mathbf{x}_{t_i}, t_i, c)\|_2^2\right], \tag{3}$$

where $w(t_i)$ is a timestep-dependent weight. When the student and teacher models share the same per-step covariance, this velocity-matching loss is equivalent to minimizing the per-step KL divergence between the two models (omit the time-dependent coefficient):

$$D_{\mathrm{KL}}(\pi_\theta(\cdot \mid \mathbf{x}_{t_i}) \,\|\, \pi_\phi(\cdot \mid \mathbf{x}_{t_i})) \propto \|v_\theta - v_\phi\|_2^2 . \tag{4}$$

3.2 Observation

Suppose that K reward-specialized teachers encode capabilities such as compositional fidelity, text rendering, and visual aesthetics. To mitigate cross-teacher supervision conflicts, existing multi-teacher OPD methods partition the training prompts into non-overlapping subsets $\{\mathcal{D}_k\}_{k=1}^{K}$ and assign each subset to one teacher:

$$v^\star = v_{\phi_k}, \qquad k = \rho(c) \;\text{ for }\; c \in \mathcal{D}_k, \tag{5}$$

where ρ is the fixed task-based routing rule and v_{ϕ_k} denotes the k-th teacher. The selected index k determines the supervision target but is not provided to the student, forcing it to infer the intended capability solely from c and thereby entangling prompt semantics with capability activation.

This design leads to two limitations. First, capability activation is sensitive to linguistic variation: when the wording or style of a prompt deviates from the routed training subsets, the prompt may activate a different expert behavior even if its visual intent is nearly identical (Fig. 1(c) and Fig. 2). This resembles the linguistic hacking observed in reward-based post-training Wang et al. (2026), but is more pronounced here because a perturbation can switch which capability is activated rather than only changing its strength. Second, semantic routing fixes both capability selection and activation strength in the prompt, so users cannot adjust the strength of a single capability or invoke multiple capabilities together for the same prompt. These limitations arise from entangling semantic content with capability intent and motivate an explicit capability-conditioning mechanism that decouples *what to generate* from *which capabilities to activate* and *at what strength*, so that capability control no longer depends on prompt wording and users can combine and adjust capabilities at inference time.

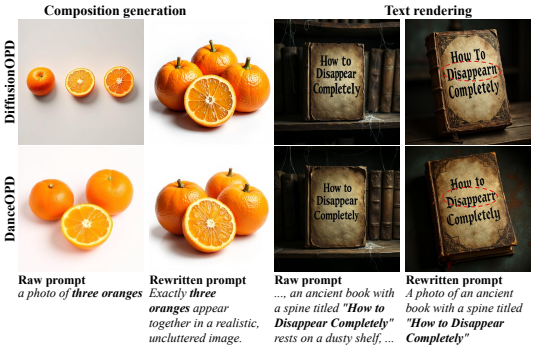

Figure 2: Visualization under prompt rewriting.

3.3 Joint-Anchored Explicit Capability Field

We parameterize capability intent by $\boldsymbol{\lambda} \in [0,1]^K$, where λ_i continuously controls capability i, and multiple nonzero entries indicate a request for capability combination. The student $v_\theta(\mathbf{x}_t, t, c, \boldsymbol{\lambda})$ receives the coordinate through a small projection added to its global conditioning, i.e.,

$$\mathbf{h}(t, c, \boldsymbol{\lambda}) = \mathbf{h}(t, c) + W_2\,\mathrm{SiLU}(W_1\boldsymbol{\lambda}). \tag{6}$$

The projection is bias-free, and W_2 is initialized to zero. This separates capability control from the text prompt without changing the text encoder or transformer blocks.

Let v_0 denote the frozen base model without post-training, v_i the expert specialized for capability i, and v_{ij} the joint teacher obtained by jointly optimizing the objectives for capabilities i and j during post-training. Consider a capability coordinate $\boldsymbol{\lambda}$ that is nonzero only on a supported pair (i, j). The corresponding target velocity is defined by the piecewise-linear capability field

$$v^*(\boldsymbol{\lambda}) = (1 - \lambda_i - \lambda_j + \lambda_{\min})v_0 + (\lambda_i - \lambda_{\min})v_i + (\lambda_j - \lambda_{\min})v_j + \lambda_{\min}v_{ij}, \tag{7}$$

where $\lambda_{\min} = \min(\lambda_i, \lambda_j)$ and, all velocities at the same student-visited state $(\mathbf{x}_t, t, c)$, omitted for clarity. The overlap $\lambda_{\min}$ of the two requests goes to the joint teacher, and each single expert covers only the excess of its own coordinate. Specifically, the corner $(\lambda_i, \lambda_j) = (0, 0)$ recovers the base model, $(1, 0)$ and $(0, 1)$ recover the two individual experts, and $(1, 1)$ recovers the joint teacher (Fig. 1(a)). Anchored at these four corners, the field is continuous and leaves each teacher exactly reachable from any direction, so composing capabilities does not degrade individual performance. More broadly, Eq. 7 is not tied to a specific pair but provides a general rule for composing any two interacting capabilities, which effectively alleviates conflicts between capabilities. The joint teacher v_{ij} contributes to the target only when both coordinates are nonzero, since its coefficient $\lambda_{\min}$ vanishes otherwise. At an axis-aligned coordinate such as $(\lambda_i, 0)$, the target reduces to interpolation between the base model and a single expert, $v^* = (1 - \lambda_i)v_0 + \lambda_i v_i$, and single-capability supervision never involves a joint teacher. The sampling region of each pair is therefore chosen according to its available anchors. For a pair with a joint teacher, coordinates are sampled over the full unit square, so that combined activations are also supervised. For a pair without a joint teacher, only axis-aligned coordinates are sampled, and the joint term is never used.

The training objective adopts the standard OPD loss, with both the student and target velocity conditioned on the capability coordinates, i.e.,

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}\Big[\|v_\theta(\mathbf{x}_t, t, c, \boldsymbol{\lambda}) - v^*(\mathbf{x}_t, t, c, \boldsymbol{\lambda})\|_2^2\Big]. \tag{8}$$

As the target velocity varies with $\boldsymbol{\lambda}$, each requested capability configuration is matched with its unique target, thus the same prompt can be supervised by different teachers or their combinations under different capability coordinates, without introducing inconsistent targets for the same input.

3.4 Capability Landscape Profiling

The value $\lambda_i = 1$ marks the training anchor of capability i, but does not necessarily yield the strongest student response. We therefore profile the reward landscape of the learned capability field to identify the coordinate with the highest mean reward and measure how often each coordinate is optimal across prompt–noise pairs. Given a capability set S, we fix all coordinates outside S to zero and construct a probe grid $\Lambda^S_{\mathrm{probe}} = \Lambda^S_{\mathrm{in}} \cup \Lambda^S_{\mathrm{ext}}$, covering the training range and a bounded extrapolation region. Every coordinate is evaluated on the same calibration bank $\mathcal{B}^S_{\mathrm{cal}} = \{(c_m, z_m)\}_{m=1}^M$, where c_m is a training prompt and z_m is its fixed initial noise. The resulting reward records are

$$r_m(\boldsymbol{\lambda}) = q_S(G_\theta(z_m, c_m, \boldsymbol{\lambda})), \quad m = 1, \ldots, M, \quad \boldsymbol{\lambda} \in \Lambda^S_{\mathrm{probe}}, \tag{9}$$

where q_S is the corresponding reward function for capability set S (can be a weighted objective under multiple capabilities), and $G_\theta(\cdot)$ is the image generator. Reusing the same prompt–noise pairs ensures that only the capability coordinates vary across evaluations. From these records, we compute the mean reward and the empirical frequency of optimality at each coordinate:

$$\bar{r}_S(\boldsymbol{\lambda}) = \frac{1}{M}\sum_{m=1}^{M} r_m(\boldsymbol{\lambda}), \qquad \widehat{P}_S(\boldsymbol{\lambda}) = \frac{1}{M}\sum_{m=1}^{M} \frac{\mathbf{1}\{\boldsymbol{\lambda} \in \Lambda^\star_m\}}{|\Lambda^\star_m|}, \tag{10}$$

where $\bar{r}_S(\boldsymbol{\lambda})$ is the average reward of coordinate $\boldsymbol{\lambda}$ over the M pairs, $\widehat{P}_S(\boldsymbol{\lambda})$ is the fraction of pairs for which $\boldsymbol{\lambda}$ is optimal, with ties split evenly among the optimal coordinates, and $\Lambda^\star_m = \arg\max_{\boldsymbol{\lambda} \in \Lambda^S_{\mathrm{probe}}} r_m(\boldsymbol{\lambda})$ is the set of optimal coordinates for the prompt-noise pair (c_m, z_m). These two statistics measure different things: a coordinate that is best on many pairs can still score poorly on the rest ($\widehat{P}_S$ can be high while $\bar{r}_S$ stays moderate), and a coordinate can also score well on every pair without ever being the best (the highest $\bar{r}_S$ can correspond to low $\widehat{P}_S$). We therefore define:

$$\widehat{\boldsymbol{\lambda}}^S_{\mathrm{peak}} = \arg\max_{\boldsymbol{\lambda} \in \Lambda^S_{\mathrm{probe}}} \bar{r}_S(\boldsymbol{\lambda}), \qquad \Lambda^S_{\mathrm{search}} = \Big\{\boldsymbol{\lambda} \in \Lambda^S_{\mathrm{probe}} : \widehat{P}_S(\boldsymbol{\lambda}) > 0\Big\}, \tag{11}$$

where $\widehat{\boldsymbol{\lambda}}^S_{\mathrm{peak}}$ is the recommended coordinate, and $\Lambda^S_{\mathrm{search}}$ collects every coordinate that is optimal for at least one prompt-noise pair, offering the candidate set for test-time search. Since $\widehat{P}_S$ sums to one and is nonzero exactly on $\Lambda^S_{\mathrm{search}}$, it can be used directly as the sampling distribution for test-time scaling. The performance gain of coordinate extrapolation can thus be quantified as follows:

$$\Delta^S_{\mathrm{ext}} = \max_{\boldsymbol{\lambda} \in \Lambda^S_{\mathrm{ext}}} \bar{r}_S(\boldsymbol{\lambda}) - \max_{\boldsymbol{\lambda} \in \Lambda^S_{\mathrm{in}}} \bar{r}_S(\boldsymbol{\lambda}). \tag{12}$$

3.5 TEST-TIME SCALING OVER CAPABILITY COORDINATES

Although the default coordinate $\widehat{\boldsymbol{\lambda}}^S_{\text{peak}}$ achieves the highest mean reward during profiling, the optimal coordinate can vary across prompt–noise pairs. We therefore perform test-time search over capability coordinates to improve generation quality for a given initial noise. Specifically, we draw N coordinates from $\Lambda^S_{\text{search}}$ without replacement, with probability proportional to $\widehat{P}_S$. Each coordinate produces a candidate record $(\mathbf{x}_n, r_n, \boldsymbol{\lambda}_n)$:

$$\mathcal{C}_N = \{(\mathbf{x}_n, r_n, \boldsymbol{\lambda}_n)\}_{n=1}^N, \qquad \mathbf{x}_n = G_\theta(z, c, \boldsymbol{\lambda}_n), \quad r_n = q_S(\mathbf{x}_n), \tag{13}$$

where the same initial noise z is used for all candidates, so that only the capability coordinates vary. We then select the record with the highest reward:

$$(\mathbf{x}^*, r^*, \boldsymbol{\lambda}^*) \in \arg \max_{(\mathbf{x}, r, \boldsymbol{\lambda}) \in \mathcal{C}_N} r. \tag{14}$$

Such a search explores different capability configurations for the same initial noise, allowing higher-reward outputs to be obtained without resampling random seeds.

4 EXPERIMENTS

Tasks and datasets. Following prior work (Fang et al., 2026), we investigate three capabilities: compositional generation, text rendering, and visual aesthetics. For compositional generation and text rendering, we use the training and test splits released by Flow-GRPO (Liu et al., 2025a) for GenEval (Ghosh et al., 2023) and OCR, respectively; for visual aesthetics, prompts are drawn from the HPD (Wu et al., 2023) dataset. Compositional generation and text rendering are evaluated with the GenEval and OCR rewards, while the aesthetic objective combines HPSv3 (Ma et al., 2025), CLIP (Radford et al., 2021), and PickScore (Kirstain et al., 2023) with equal weights 1:1:1. For capability landscape profiling, the calibration bank is drawn from the training split with 200 prompts per capability set, so that coordinate selection never touches testing data.

Teacher construction. We train five teacher models using Flow-GRPO-Fast (Liu et al., 2025a). The three single-capability teachers are trained on the corresponding prompts and rewards above. We additionally train two joint-capability teachers, GenEval+aesthetics on GenEval prompts and OCR+aesthetics on OCR prompts, where every generated sample is evaluated by the task reward together with HPSv3, CLIP, and PickScore at a coefficient ratio of 3:1:1:1. We do not build a GenEval+OCR teacher because the two prompt sets have incompatible formats: GenEval prompts are template-based scene descriptions, whereas OCR prompts must contain a quoted string to render, so no natural prompt requires both capabilities. Aesthetics imposes no constraint on prompt format and thus pairs with either task; the joint teachers directly learn the combination of task correctness and visual quality, providing anchors for coordinates where both capabilities are activated.

Baselines. We use FLUX.1-dev (Labs, 2024) as the backbone for all teacher and student models and apply LoRA with $r = 64$ and $\alpha = 128$. Rollout and distillation use 10 sampling steps; evaluation uses 28 steps, and all images are generated at a resolution of 512×512. We compare CapField-OPD with the three single-capability teachers, a multi-task teacher trained through multi-objective reinforcement learning (Xue et al., 2025b), DiffusionOPD (Li et al., 2026c), and DanceOPD (Zhou et al., 2026a); all OPD methods use the same teacher models for a fair comparison. More implementation details can be found in the Appendix.

4.1 MAIN RESULTS

Tab. 1 presents the main quantitative comparison. The single-capability teachers perform strongly on their target tasks, while the joint teachers provide better balance between correctness and visual quality. Multi-task GRPO and existing OPD methods consolidate these capabilities into one model with a single operating point, and adding joint teachers (Single+Joint) lifts several aesthetic metrics but consistently lowers the GenEval and OCR scores. Without teacher identity as input, this supervision shifts the learned compromise rather than creating separately accessible capability modes. In comparison, the two CapField-OPD rows come from the same student model and differ only in capability coordinates. Single-capability coordinates achieve the best GenEval and OCR scores

Table 1: Quantitative comparison. *Single-only* and *Single+Joint* distill from the three single-capability teachers without and with the two joint-capability teachers, respectively. The two CapField-OPD modes share one student model and differ only in their inference coordinates. Bold and underlined values denote the best and second-best results among unified models. Since aesthetic prompts contain no composition target or text-rendering target, the joint mode does not apply, and the corresponding entries are marked "–".

Method	Composition-task prompts				Text-rendering prompts				Aesthetic prompts		
	GenEval	HPSv3	CLIP	PickScore	OCR	HPSv3	CLIP	PickScore	HPSv3	CLIP	PickScore
FLUX.1-dev	0.6616	8.65	0.3966	23.40	0.5735	13.00	0.4466	22.91	13.28	0.3868	22.58
GenEval teacher	0.9417	9.16	0.4092	23.27	0.6170	13.43	0.4585	22.97	13.49	0.3946	22.47
OCR teacher	0.7166	8.90	0.4030	23.59	0.9417	12.74	0.4567	22.88	13.57	0.3834	22.64
Aesthetic teacher	0.3228	10.62	0.4117	24.04	0.4824	15.14	0.4688	23.95	15.23	0.4181	23.68
GenEval+Aesthetic teacher	0.9253	11.75	0.4210	24.11	0.6235	14.74	0.4525	23.32	14.65	0.3968	22.71
OCR+Aesthetic teacher	0.6041	9.46	0.4019	23.59	0.9031	14.40	0.4631	23.53	14.50	0.3945	23.06
Multi-task GRPO	0.8608	9.43	<u>0.4196</u>	<u>23.89</u>	0.9369	13.20	**0.4638**	23.19	14.32	0.4024	23.15
DiffusionOPD (Single-only)	<u>0.9514</u>	8.37	0.4167	23.02	<u>0.9409</u>	12.76	0.4574	22.90	15.25	0.4174	**23.72**
DiffusionOPD (Single+Joint)	0.9434	10.96	0.4188	23.58	0.9229	13.86	<u>0.4632</u>	<u>23.36</u>	15.25	<u>0.4180</u>	23.70
DanceOPD (Single-only)	0.9239	7.72	0.4092	22.97	0.9289	12.87	0.4579	22.95	15.19	0.4156	23.64
DanceOPD (Single+Joint)	0.9236	<u>11.19</u>	0.4123	23.73	0.8875	**14.09**	0.4604	**23.44**	<u>15.27</u>	0.4162	23.70
CapField-OPD (Single mode)	**0.9531**	8.50	0.4170	23.02	**0.9592**	11.96	0.4439	22.51	**15.32**	**0.4182**	<u>23.71</u>
CapField-OPD (Joint mode)	0.9322	**11.76**	**0.4201**	**23.94**	0.9256	<u>14.08</u>	0.4538	23.32	–	–	–

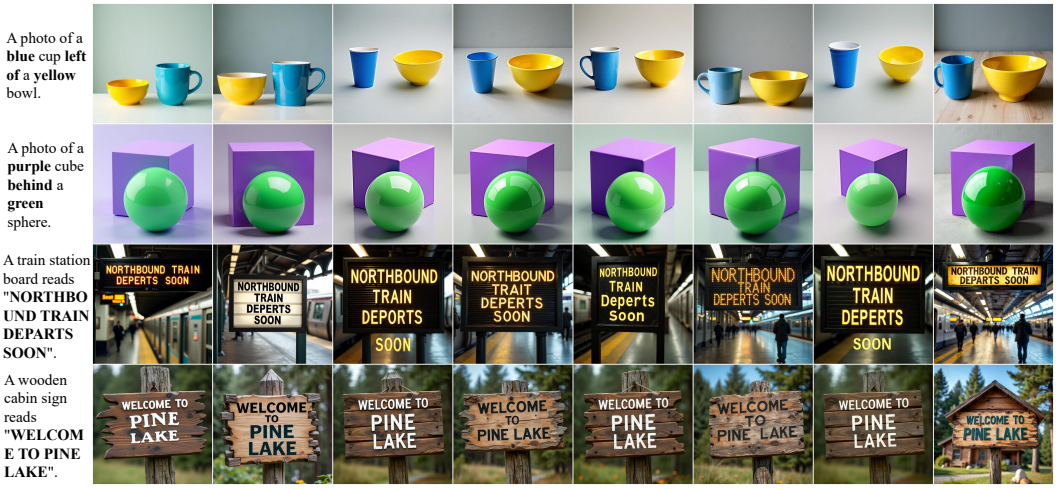

Figure 3: Visual demonstration of different methods, from left to right: FLUX.1-dev, Multi-task GRPO, DiffusionOPD(Single-only), DiffusionOPD(Single+Joint), DanceOPD(Single-only), DanceOPD(Single+Joint), CapField-OPD(Single mode), and CapField-OPD(Joint mode).

overall and slightly exceed the aesthetic teacher on all three aesthetic metrics, and some of these coordinates lie beyond the training anchors, so the anchors do not limit the learned field. Joint-capability coordinates give better-balanced operating points while retaining strong task accuracy. CapField-OPD thus supports capability extrapolation and direct switching between specialized and joint behaviors in one model, without retraining. Fig. 3 shows the same trend qualitatively: in single mode, CapField-OPD matches the strongest baseline on text rendering, and joint mode improves visual quality while keeping the text correct.

4.2 Robustness to Prompt Rewriting

We rewrite the benchmark prompts with GPT-5.6 (OpenAI, 2026) while preserving their task content and evaluate all methods without retraining. As shown in Tab. 2, CapField-OPD achieves the best result on all three metrics, while the compared OPD methods degrade significantly, especially in the GenEval and OCR tasks. This is because these methods infer the capability from prompt semantics, so the rewriting operator perturbs the inferred capability. CapField-OPD instead invokes it through an explicit coordinate that does not change with the wording, decoupling control from prompt phrasing: the learned capabilities transfer to new prompt types and styles.

Table 2: Robustness evaluation under semantics-preserving prompt rewriting.

Metric	FLUX.1-dev	Multi-task GRPO	DiffusionOPD		DanceOPD		CapField-OPD Single mode
			Single	Single+Joint	Single	Single+Joint	
GenEval ↑	0.6550	0.8461	0.6241	0.6132	0.6838	0.5560	**0.9267**
OCR ↑	0.6115	0.9242	0.8662	0.8718	0.8879	0.8097	**0.9677**
HPSv3 ↑	13.46	14.45	15.13	15.15	15.08	15.17	**15.28**



Figure 4: Illustration of continuous capability control at inference time.

4.3 Continuous Capability Control

As shown in Fig. 4, sweeping the capability coordinates with the prompt and initial noise fixed produces smooth transitions between task-oriented and aesthetic behaviors. Moving from GenEval toward aesthetics gradually enriches scene details, whereas moving from OCR toward aesthetics changes lighting and visual style while text fidelity is strongest near the OCR endpoint. The main subjects and overall content remain recognizable throughout each sweep, showing that the coordinates adjust the relative capability emphasis without abrupt behavior changes.

4.4 Test-Time Scaling over Capability Coordinates

We compare coordinate-based scaling with conventional seed-based scaling under the same generation budget $N \in \{3, 6, 9\}$. Seed-based scaling fixes the capability configuration and resamples the initial noise, while coordinate-based scaling fixes the noise and searches over capability coordinates. As shown in Fig. 5, coordinate-based scaling improves the reward steadily as the budget grows and matches seed-based scaling across the three reward objectives, with a difference of about 0.5%. Moreover, seed-based scaling gains reward by resampling the noise, so a higher-reward candidate often comes with a different scene layout or visual appearance. In comparison, coordinate-based scaling keeps the noise fixed and adjusts only capability strengths, so task errors such as incorrect text or wrong object counts are corrected while the original scene is largely preserved. Coordinate search is therefore preferable when the initial generation is largely satisfactory and only minor capability defects remain to be fixed.

4.5 Ablation Studies

Effect of explicit capability coordinates. Tab. 1 illustrates the limitation of consolidating teacher behaviors into a single operating point. For DiffusionOPD and DanceOPD, adding joint teachers improves aesthetic metrics but reduces task accuracy, because the student receives no capability signal and one prompt can correspond to several plausible targets. CapField-OPD resolves this ambiguity by making capability intent explicit through coordinates, enabling a single student to support both specialized and joint behaviors.

Effect of joint teachers. Tab. 3 isolates the contribution of joint anchors under the same conditioning architecture. Adding joint teachers increases GenEval from 0.7723 to 0.9322 and OCR from 0.8716 to 0.9256 under joint activation, while improving all three aesthetic metrics for both capability pairs. Thus, single-capability anchors alone do not fully determine the behavior required at joint coordinates; joint teachers directly anchor these combined operating points.

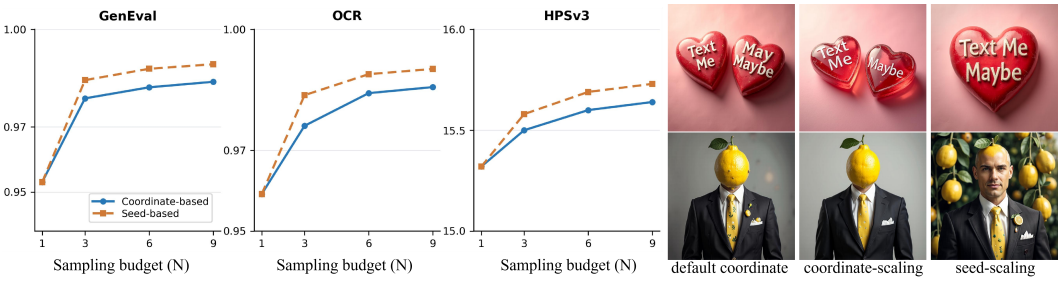

Figure 5: Demonstration of coordinate-based and seed-based test-time scaling.

Table 3: Effect of joint-teacher anchors at jointly activated tasks.

Training anchors	Composition + Aesthetics				Text Rendering + Aesthetics			
	GenEval ↑	HPSv3 ↑	CLIP ↑	PickScore ↑	OCR ↑	HPSv3 ↑	CLIP ↑	PickScore ↑
Base + single	0.7723	9.38	0.4034	22.23	0.8716	13.50	0.4481	22.84
Base + single + joint	**0.9322**	**11.76**	**0.4201**	**23.94**	**0.9256**	**14.08**	**0.4538**	**23.32**

Table 4: Effect of coordinate extrapolation. Each $\hat{\lambda}^{S}_{\text{peak}}$ (Eq. 11) reports the full selected coordinate over (GenEval, OCR, aesthetics), with S being the single capability of the corresponding metric.

Coordinate selection	Composition		Text Rendering		Aesthetics			
	$\hat{\lambda}^{S}_{\text{peak}}$	GenEval ↑	$\hat{\lambda}^{S}_{\text{peak}}$	OCR ↑	$\hat{\lambda}^{S}_{\text{peak}}$	HPSv3 ↑	CLIP ↑	PickScore ↑
Teacher model	/	0.9417	/	0.9417	/	15.23	0.4181	23.68
In-range profiling	(1.0, 0.0, 0.0)	0.9487	(0.0, 1.0, 0.0)	0.9390	(0.0, 0.0, 1.0)	15.25	0.4171	23.70
Expanded profiling	(1.05, 0.0, 0.0)	**0.9531**	(0.0, 1.5, 0.0)	**0.9592**	(0.0, 0.0, 1.3)	**15.32**	**0.4182**	**23.71**

Effect of coordinate extrapolation. As shown in Tab. 4, expanded profiling improves all metrics over in-range profiling, and the selected coordinates lie beyond the training range. This improvement arises from the continuity of the learned field: it maps coordinate changes to response changes, so the trend observed between anchors extends past them. The training anchors therefore define reference states rather than the best inference settings of the learned field.

4.6 Limitation

Although the training target is constructed by fusing multiple teacher outputs, these outputs do not participate in rollout and require no gradients, so the extra cost is limited. In practice, compared to DiffusionOPD under the same backbone, batch size, and sampling steps, the per-step training time increases only from 11 s to 12 s (about 9%), which comes from the additional frozen-teacher forward passes used to build the joint-anchored target. Additionally, the proposed method builds the student by consolidating the capabilities of the teachers. If a teacher trained in the RL stage suffers from problems such as reward hacking, the student may inherit similar behaviors from its supervision.

5 Conclusion

In this work, we present CapField-OPD, an on-policy distillation framework that learns a continuous capability field over explicit capability coordinates, where reward-specialized teachers serve as anchors. Existing multi-teacher OPD methods infer the desired capability from prompt content, which binds capability control to prompt wording and prevents users from adjusting it at inference time. By conditioning the student on these coordinates, CapField-OPD decouples capability control from prompt semantics. The student thus matches or exceeds every anchor on its target task, and the requested capability stays active under prompt rewriting. The coordinates can also extend beyond the training anchors, which serve as reference states rather than hard performance limits. More broadly, explicit capability coordinates offer a way to build generative models whose behaviors are specified by users rather than inferred from prompts.

AI USE STATEMENT

Generative AI tools were used solely for minor language editing and polishing. All AI-assisted text was reviewed and revised by the authors. The authors take full responsibility for the content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies capability consolidation and control in text-to-image flow models using publicly available models and evaluation benchmarks, and does not involve human subjects or the collection of personal data. CapField-OPD does not introduce risks specific to its distillation procedure beyond those generally associated with text-to-image generation, such as biased, misleading, or harmful outputs. Moreover, because the student learns from reward-specialized teachers, it may inherit biases or reward-hacking behaviors present in those teachers. Appropriate content safeguards and careful evaluation are therefore necessary when deploying such models.

REPRODUCIBILITY STATEMENT

To support reproducible research, we provide detailed hyperparameter settings and implementation details in the appendix. An anonymized implementation of CapField-OPD is also included in the supplementary material. The complete source code and configurations will be released upon acceptance. We hope this release facilitates future research on explicit capability control and multi-teacher on-policy distillation for flow-based generative models.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Tianyi Wei, Xiaohang Zhan, Jiaqi Wang, Tong Wu, Xingang Pan, et al. From sparse to dense: Multi-view grpo for flow models via augmented condition space. In *European Conference on Computer Vision*, pp. 371–389. Springer, 2026.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16185–16195. IEEE, 2025.

Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. Twinflow: Realizing one-step generation on large models with self-adversarial flows. *arXiv preprint arXiv:2512.05150*, 2025.

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

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.

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.

Siming Fu, Zheming Fu, Ruizhe He, Hualiang Wang, Jie Huang, Xiaoxiao Ma, Mingchen Zhong, Weihu Huang, Xiaoxuan He, and Haojun Xu. Any-opd: Heterogeneous on-policy distillation for flow-matching models via representation-space bridging. *arXiv preprint arXiv:2608.03316*, 2026.

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.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Bingnan Li, Haozhe Wang, Haozhong Xiong, Fangtai Wu, Jinpeng Yu, Yang Shi, Jiaming Liu, and Ruihua Huang. Rethinking classifier-free guidance in on-policy diffusion distillation. *arXiv preprint arXiv:2607.24731*, 2026a.

Jiaming Li, Chenyu Zhu, Nanxi Yi, Youjun Bao, Li Sun, Quanying Lv, Xiang Fang, Daizong Liu, Jianjun Li, Kun He, et al. Tmpo: Trajectory matching policy optimization for diverse and efficient diffusion alignment. *arXiv preprint arXiv:2605.10983*, 2026b.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, et al. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

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

Mingfeng Lin, Chengfei Cai, Lin Xu, Yuxiang Wei, and Liang Han. Dreopd: Degraded-reference extrapolative on-policy distillation for flow-matching models. *arXiv preprint arXiv:2608.09233*, 2026.

Pengyang Ling, Jiazi Bu, Yujie Zhou, Yibin Wang, Zhenyu Hu, Zihan Zhang, Yi Jin, Huaian Chen, and Yuhang Zang. Pave-grpo: Beyond instantaneous guidance through principled average velocity decomposition. *arXiv preprint arXiv:2606.01636*, 2026.

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

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. *arXiv preprint arXiv:2505.05470*, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025b.

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.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025.

OpenAI. GPT-5.6 System Card, 2026. URL `https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf`.

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

Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

Fu-Yun Wang, Han Zhang, Michael Gharbi, Hongsheng Li, and Taesung Park. Promptrl: Prompt matters in rl for flow-based image generation. *arXiv preprint arXiv:2602.01382*, 2026.

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

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8941–8951. IEEE, 2024.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

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

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

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

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained grpo for precise preference alignment in flow models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20045–20054, June 2026b.

A APPENDIX

This appendix provides the training pseudocode of CapField-OPD (Sec. A.1), the full hyperparameter configuration together with the capability curves during distillation (Sec. A.2), and additional visualizations of coordinate-based strength control (Sec. A.3).

A.1 PSEUDOCODE

Algorithm 1 summarizes CapField-OPD training. The single-capability and joint teachers are obtained by reward-based post-training and then frozen, and they supervise the student only at student-visited states. Each prompt c is paired with a set $\Lambda(c)$ of matching capability coordinates, and all coordinate sampling is tied to this set. Each rollout is conditioned on a coordinate $\boldsymbol{\lambda}^{\mathrm{roll}}$ drawn from $\Lambda(c)$, and the capability coordinates can be different when computing OPD loss. A single rollout thus supervises many coordinates, so the rollout cost does not scale with the number of supervised coordinates. For pairs without a joint teacher, both the rollout and the supervised coordinates are restricted to axis-aligned ones (Sec. 3.3).

Algorithm 1 CapField-OPD Training

Input: Frozen base teacher v_0, single and joint teachers $\{v_i\}, \{v_{ij}\}$; prompt datasets $\{\mathcal{D}_k\}$, where each prompt c is paired with its matching capability coordinates $\Lambda(c)$; noise schedule $\{t_r\}_{r=0}^{S}$.
Output: Student $v_\theta(\mathbf{x}, t, c, \boldsymbol{\lambda})$.
Initialize v_θ from v_0 with a trainable LoRA; add the coordinate conditioner (Eq. 6) with $W_2 \leftarrow \mathbf{0}$.
for each training iteration **do**
  Sample a balanced batch of prompts $c \sim \mathcal{D}_k$; for each prompt, draw a rollout coordinate $\boldsymbol{\lambda}^{\mathrm{roll}} \sim \Lambda(c)$.
  Roll out the student on each $(c, \boldsymbol{\lambda}^{\mathrm{roll}})$ to obtain on-policy trajectories $\{\mathbf{x}_{t_r}\}_{r=0}^{S}$. ▷ no gradient
  for each intermediate state $(\mathbf{x}_{t_r}, t_r, c)$ **do**
    Draw a capability coordinate $\boldsymbol{\lambda} \sim \Lambda(c)$ for loss computation.
    Compute the teacher target $\mathbf{v}^\star$ via Eq. 7 at $(\mathbf{x}_{t_r}, t_r, c)$. ▷ no gradient
    Accumulate the squared error $\|v_\theta(\mathbf{x}_{t_r}, t_r, c, \boldsymbol{\lambda}) - \mathbf{v}^\star\|_2^2$ into $\mathcal{L}_{\mathrm{OPD}}$.
  end for
  Update the LoRA and the coordinate conditioner based on the mean of $\mathcal{L}_{\mathrm{OPD}}$.
end for

A.2 HYPERPARAMETER CONFIGURATION AND CAPABILITY CURVES

Table 5 lists the hyperparameter settings used for CapField-OPD training. Fig. 6 reports the validation reward of each capability over distillation steps, evaluated every 200 steps on validation prompts per capability set, in which all the capabilities improve steadily during training.

Table 5: Hyperparameter settings used for CapField-OPD.

Parameter	Value	Parameter	Value
Random seed	42	Learning rate	2×10^{-4}
Optimizer	AdamW	Weight decay	1×10^{-4}
AdamW β_1, β_2	0.9, 0.999	AdamW ϵ	1×10^{-8}
LR scheduler	Constant	Text length (CLIP / T5)	77 / 256
LoRA rank / α	64 / 128	Cond. dimensions	$3\rightarrow256\rightarrow3072$
Batch size	8	Number of GPUs	8
Mixed precision	bfloat16	Max. grad norm	1.0
Denoising steps	10	Train time indices	$0, \ldots, 9$
Resolution	512×512	Guidance scale	4.5
Loss target	Velocity	Timestep aggregation	Mean
EMA decay	0.9	EMA update interval	8 steps
Calibration size M	200	Probe grid spacing	0.05
Extrapolation bound	1.5	Search budget N	$\{3, 6, 9\}$

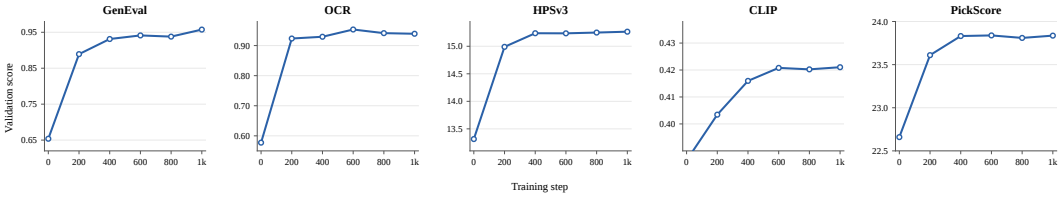

Figure 6: Capability curves during CapField-OPD training.

A.3 Additional Visualizations of Capability Control

Fig. 7 and Fig. 8 represent more visual results of coordinate sweeping at inference time for the two supported pairs, (GenEval, aesthetics) and (OCR, aesthetics). Each row fixes the prompt and the initial noise and varies only the coordinates, so the differences within a row come from the capability field rather than the seed. For a given prompt-noise pair, sweeping the coordinate moves the output smoothly along the corresponding capability axis while the composition stays close to the shared starting point. The control behavior is therefore a property of the field itself, not of a particular capability pair or seed.



Figure 7: Additional coordinate sweeps at inference time from GenEval to aesthetics.

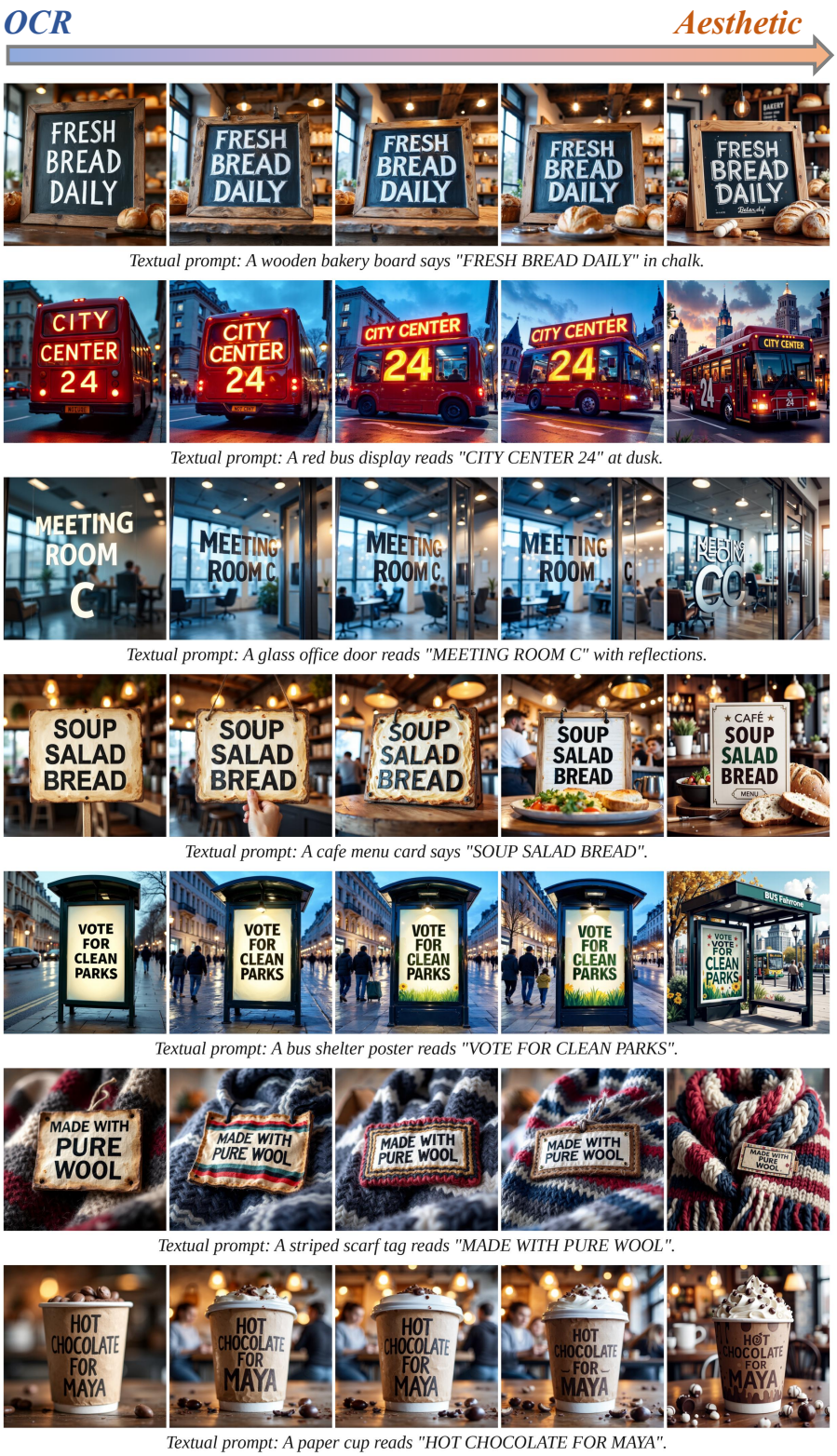

Figure 8: Additional coordinate sweeps at inference time from OCR to aesthetics.