SCORED CONSISTENCY DISTILLATION: CHOOSING *Which* TEACHER TRAJECTORY TO IMITATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step distillation trains a student to imitate a teacher's sampling trajectory, but which trajectory the teacher produces is decided by the seed. For a single caption a teacher generates outcomes that differ widely in how faithfully they realise the prompt, and standard distillation imitates one of them at random, inheriting the teacher's compositional failures as readily as its successes. We show that choosing the trajectory to imitate, and adding a reward on the student's own clean-latent estimates, improves *compositional alignment* with low online overhead. For each caption the frozen teacher samples several candidates; we select the one whose mean-pooled DINOv2 patch embedding is closest to the caption's real photograph, and distil on that trajectory alone. A lightweight projector maps the student's terminal latent directly into DINO space, so the same score applies to the student's own predictions. The projector tracks the true score on the student's recent predictions, and the two are updated on separate steps, which prevents direct co-adaptation between the reward and the policy it scores. On SD3.5-Medium distilled to four steps, the method improves T2I-CompBench++ from 0.4463 to 0.5036 and GenEval2 from 21.3 to 23.7, with the largest gains on the spatial categories, followed by texture and numeracy. The four-step student surpasses the ten-step teacher it was distilled from and matches the teacher at its recommended 40-step setting, at a fraction of the inference budget.

1 INTRODUCTION

Denoising generative models are now the dominant paradigm for text-to-image synthesis. Diffusion models (Ho et al., 2020; Song et al., 2021) and their rectified-flow reformulation (Lipman et al., 2023; Liu et al., 2023) underpin the current generation of large open systems, from latent UNet architectures (Rombach et al., 2022; Podell et al., 2024) to multimodal diffusion transformers (Peebles & Xie, 2023; Esser et al., 2024). Their central practical drawback is that sampling traverses a trajectory of tens of network evaluations, which is prohibitive for interactive or high-throughput use, and a large literature has grown around decreasing that cost (Ma et al., 2025). The most effective solution is *distillation*. A frozen many-step teacher supplies a sampling trajectory, and a student learns to reproduce the teacher's clean-data estimates from fewer and noisier inputs. Progressive distillation repeatedly halves the sampler steps (Salimans & Ho, 2022), consistency models enforce self-consistency along the probability-flow ODE to reach few- or single-step generation (Song et al., 2023; Song & Dhariwal, 2024; Luo et al., 2023; Kim et al., 2024), and guidance distillation folds classifier-free guidance into the student so that two forward passes become one (Meng et al., 2023).

A more recent line replaces pointwise trajectory imitation with distributional, adversarial or score-based objectives (Yin et al., 2024b;a; Sauer et al., 2024; Luo et al., 2024; Zhou et al., 2024), and few-step students built this way now rival or exceed the multi-step models they came from (Jiang et al., 2026; Zhang et al., 2026). To assess what these models can actually render, Huang et al. (2023) and Ghosh et al. (2023) introduced compositional benchmarks, which test what the structure of a prompt specifies beyond its individual words, namely which objects appear and in what number, which attribute binds to which object, and how the objects are arranged relative to one another.

Recent studies treat distillation mainly as a compression problem. The consistency objective measures only agreement with the teacher output, so the student inherits the compositional content of that output, and this dependence has received little attention. This framing takes the supervision

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.

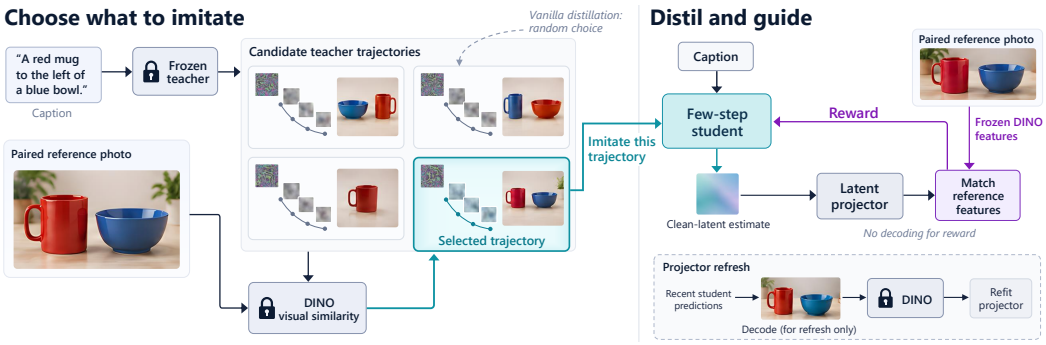

Figure 1: **Scored consistency distillation.** Left: for each caption the frozen teacher samples several candidate trajectories, and the candidate whose final image is closest to the caption's paired photograph in DINOv2 feature space becomes the distillation target, where vanilla distillation takes a random candidate. Right: the few-step student imitates the selected trajectory, and a latent projector scores the student's clean-latent estimate against the reference features without decoding it. The projector is periodically refit on decoded student predictions (bottom right), in steps separate from the student updates.

itself for granted. A teacher trajectory is a sample from a distribution, so re-running the teacher from a different seed on the same caption potentially yields a different image. The differences reach well beyond style. On compositional prompts the seed decides whether both named objects appear, whether a stated attribute binds to the correct noun, and whether a spatial relation holds, and these failures persist even in models trained with heavily improved captions (Betker et al., 2023). A substantial share of these failures is attributable to the initial noise rather than to the model, so the same prompt yields a correct or an incorrect composition depending on the seed alone (Guo et al., 2024; Li et al., 2025). This is why ranking a handful of candidate generations at inference time and returning the best remains such a strong baseline (Lin et al., 2024), and why fine-tuning on reward-filtered self-generated images is an effective recipe (Huang et al., 2023; Guo et al., 2026). Standard distillation leaves this variance unexploited. The student imitates a single trajectory drawn at random from the teacher's conditional sampling distribution, so a generation that binds the wrong colour contributes as much gradient as one that binds the right one, and the student is fitted to the teacher's average behaviour with its compositional errors included.

This paper asks: *If the teacher is going to generate several trajectories anyway, does it help to choose which one the student imitates?*

We introduce scored consistency distillation (Figure 1). For every training caption the frozen teacher samples four candidate trajectories from fixed seeds. We score each candidate by the cosine similarity between its mean-pooled DINOv2 (Oquab et al., 2024) patch embedding and that of the real photograph the caption was written for, and we distil on the argmax candidate alone. Selection determines which target the student imitates but never evaluates what the student itself produces. We therefore add a reward on its clean-latent estimates at the two least-noisy supervised states. A small projector maps the terminal latent into DINO patch space, which reduces the reward to one convolutional forward pass. We pretrain the projector on teacher candidates and refit it periodically against the true score on the student's recent predictions, with a replay batch that preserves its ranking of teacher candidates. The student and the projector are updated on separate steps, which keeps the reward a fixed function within each student phase.

Contributions.

- We identify trajectory choice as an unused degree of freedom in few-step distillation, and show that a text-free reference-photograph DINO score is a sufficient selection signal (§3).
- We make the same score usable as an on-policy reward through a decode-free latent projector, refreshed on the student's own predictions on a schedule that alternates with student updates (§3.3).

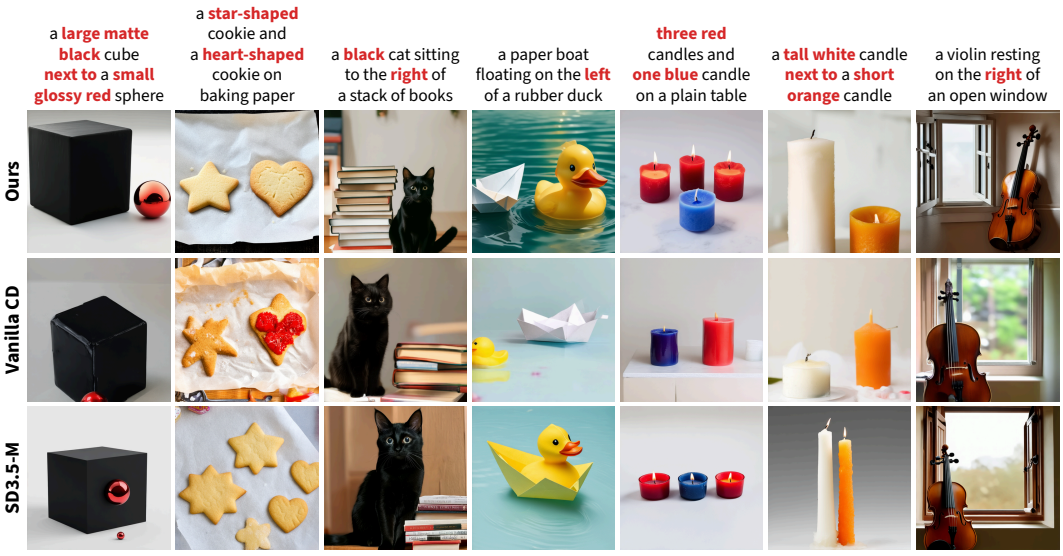

Figure 2: **Spatial-relation prompts from T2I-CompBench++**, with the relation word in red. Rows show our four-step distilled student, the vanilla consistency-distilled student at four steps, and the SD3.5-Medium base model.

- We evaluate on T2I-CompBench++, GenEval2 and text-guided image editing (§4). Against vanilla consistency distillation, the four-step student raises T2I-CompBench++ from 0.4463 to 0.5036 and GenEval2 from 21.3 to 23.7 while lowering FID from 33.16 to 26.80, and it exceeds the ten-step teacher it was distilled from (Figure 2). As an editing backbone it attains the highest ImageReward and CLIP alignment on PIE-Bench and DIV2K, above FlowEdit with SoftREPA.

2 Background

2.1 Rectified flow and few-step distillation

We work with rectified flow models (Lipman et al., 2023; Liu et al., 2023), which underlie the current generation of text-to-image transformers (Esser et al., 2024). A clean latent x_0 and noise $\epsilon \sim \mathcal{N}(0, I)$ are interpolated linearly,

$$z_\sigma = (1-\sigma)\, x_0 + \sigma\, \epsilon, \qquad \sigma \in [0,1], \tag{1}$$

and the network $v_\theta(z_\sigma, \sigma, c)$ predicts the velocity $v = \epsilon - x_0$ under text conditioning c. We index a discrete schedule so that the noise level increases with the index, $0 = \sigma_0 < \sigma_1 < \cdots < \sigma_K = 1$, and sampling walks the index downwards from K to 0 with Euler steps

$$z_{k-1} = z_k + (\sigma_{k-1} - \sigma_k)\, v_\theta(z_k, \sigma_k, c), \tag{2}$$

where the step coefficient is negative because $\sigma_{k-1} < \sigma_k$. Throughout the paper a larger index denotes a noisier state.

Two identities recur throughout the paper. The clean-latent estimate implied by a velocity at noise level σ is

$$\hat{x}_0(z_\sigma) = z_\sigma - \sigma\, v_\theta(z_\sigma, \sigma, c), \tag{3}$$

and the realised velocity between two adjacent states of an already-sampled trajectory is the finite difference $v_k = (z_k - z_{k-1})/(\sigma_k - \sigma_{k-1})$. The second identity means that a teacher's own clean estimate at step k, namely $x_{0,k} = z_k - \sigma_k v_k$, can be read off a stored trajectory without any additional forward pass.

Few-step distillation compresses this integration into a handful of steps. Writing θ for the student and taking targets from a frozen teacher, trajectory-imitation methods minimise a discrepancy between

the student's clean estimate from a noisier state and the teacher's estimate from a cleaner one,

$$\mathcal{L} \;=\; \mathbb{E}_{k,\,d}\,\mathcal{D}\Big(\hat{x}_0^{\theta}(z_{k+d}),\ \mathrm{sg}\big[\,x_{0,k}\,\big]\Big), \tag{4}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient and d is the step gap.

2.2 Self-distillation and fixed targets

Self-distillation trains a student against a target produced by the model itself or by an exponential moving average (EMA) of its parameters, as in Mean Teacher (Tarvainen & Valpola, 2017), BYOL (Grill et al., 2020) and DINO (Caron et al., 2021). The EMA smooths the noise of individual updates and gives a more stable target than the student (Tarvainen & Valpola, 2017). Because such a target depends on the parameters being trained, the objective admits collapsed solutions, which these methods avoid through asymmetry between the two branches, namely augmented views with centring and sharpening in DINO, a student-only predictor in BYOL, and a stop-gradient with a predictor in SimSiam (Chen & He, 2021). Consistency training has no teacher and avoids collapse through a boundary condition that fixes the consistency function to the identity at the minimum noise level (Song et al., 2023).

Distillation from a frozen teacher, as in equation 4, uses a target $x_{0,k}$ computed by a network whose parameters never depend on the student, so the distillation term is stable by construction. In our method the self-referential component is the reward, whose projector is refit on the student's own predictions (§3.3), and its update schedule is designed around that dependence.

3 Method

Our method changes two things about how a student is distilled. The teacher generates several candidate trajectories for each caption and we train on the one whose final image best matches the caption's real photograph in DINO feature space, and we add a reward that scores the student's own clean-latent estimates against that same photograph through a learned projector which is refreshed on the student's outputs as training proceeds. In terms of equation 4, the first change determines which trajectory $\{z_k\}$ enters the expectation and the second adds a term acting directly on $\hat{x}_0^{\theta}$. §3.1 covers candidate generation and scoring, §3.2 the distillation loss, and §3.3 the reward and its refresh schedule. Algorithm 1 in Appendix C gives the full training step.

Distillation as supervision reweighting. Selection has a direct description at the level of distributions. Let $\tau \sim p_{\mathcal{T}}(\tau \mid c)$ be a stochastic teacher trajectory for caption c, let $r(\tau) = r(\tau, x^{\mathrm{ref}})$ be the score assigned to it, and let F_r be the cumulative distribution function of $r(\tau)$ under $p_{\mathcal{T}}$. Standard trajectory distillation draws its supervision from $p_{\mathcal{T}}$ and therefore fits the teacher's average behaviour. Our method draws J independent trajectories and keeps the one of highest score. For continuous scores the selected trajectory then has density

$$p_{\mathrm{best},J}(\tau \mid c, x^{\mathrm{ref}}) \;=\; J\,p_{\mathcal{T}}(\tau \mid c)\,F_r\big(r(\tau)\big)^{J-1}, \tag{5}$$

since τ is selected when it occupies one of the J positions and the remaining $J-1$ candidates all score lower. Equation equation 5 reweights the teacher's own distribution towards its upper tail without changing the teacher, and it recovers $p_{\mathcal{T}}$ at $J = 1$.

Two properties of equation 5 matter for the method. The weight $F_r(r(\tau))^{J-1}$ depends on a trajectory's score only through its rank within the teacher distribution, so best-of-J selection is invariant to any strictly increasing transformation of r. The method therefore requires the score to order candidates correctly and does not require it to be calibrated, which is also why the projector of §3.3 is trained with a ranking objective. The reweighting is also bounded, since the divergence of $p_{\mathrm{best},J}$ from $p_{\mathcal{T}}$ grows only as $\log J$ (Appendix H).

Softmax weighting over the candidates targets a different distribution, an exponential tilt of $p_{\mathcal{T}}$ that depends on the value of the score rather than its rank, and we compare the two in Appendix H.

3.1 Candidate generation and the reference score

Candidates. For a training caption indexed by idx with paired photograph x^{ref}, the frozen teacher runs K Euler steps at classifier-free guidance w from each of J initial noises, where candidate j is

seeded deterministically by $\mathrm{idx} \cdot 1000 + j$. Deterministic seeding keeps the cost of the candidate pool low, because the cache stores only the J scalar scores and the trainer regenerates the winning trajectory from its seed when the caption is next visited, so no latents are written to disk.

Score. Let $u(\cdot)$ denote the mean over DINOv2 (Oquab et al., 2024) patch tokens with the CLS token dropped, followed by ℓ_2 normalisation. The CLS token is trained to be invariant to the crops, flips and colour jitter applied during DINO training (Caron et al., 2021), and inverting it recovers texture and semantics but not the layout of the scene (Calvo-González & Fleuret, 2026). Patch tokens are spatially organised and carry the spatial structure of the image (Singh et al., 2026), and their mean aggregates object- and attribute-level evidence from every region of the image. With I_j the decoded candidate,

$$S_j \;=\; \cos\big(u(I_j),\, u(x^{\mathrm{ref}})\big), \qquad j^\star \;=\; \arg\max_j S_j, \tag{6}$$

and we train on trajectory $j^\star$. Scoring runs once and offline, and the scorer stays frozen, so no gradient reaches it at any point during distillation. Taking the argmax implements the best-of-J rule equation 5. We also implemented the finite-temperature softmax rules, which weight every candidate by $\mathrm{softmax}(S_j/T)$ or sample one candidate from that distribution and so estimate the tilted target of Appendix H. Across the temperatures we tested, no softmax rule improved on the argmax, and exact weighting costs roughly $3\times$ the compute because it requires a student forward pass per candidate, so we use the argmax throughout (Appendix H).

The score contains no text encoder. The caption reaches it only through the photograph that the caption was written for, which acts as a single positive exemplar of what the caption should look like. This has three consequences for the method. It requires an image-caption corpus rather than captions alone, it makes the signal informative about layout, object identity and attributes while leaving it largely uninformative about relational verbs that are not properties of appearance, and it means the reference is one sample from the conditional distribution rather than its mode, so the score rewards proximity to a single valid realisation. The per-category results in §4.1 show exactly this pattern.

3.2 Distillation on the selected trajectory

Let $\{z_k\}_{k=0}^{K}$ be the teacher states of trajectory $j^\star$ and let $x_{0,k}$ be the teacher's clean estimates recovered by finite difference as in §2.1. We supervise the states whose noise level lies in a middle band, $\mathcal{W} = \{k : \sigma_k \in [\sigma_{\mathrm{lo}}, \sigma_{\mathrm{hi}}]\}$, with a random step gap $d \sim \mathcal{U}\{1, 2, 3\}$,

$$x_{0,k} = z_k - \sigma_k v_k, \qquad v_k = (z_k - z_{k-1})/(\sigma_k - \sigma_{k-1}), \tag{7}$$

$$\hat{x}^{\theta}_{0,k} = z_{k+d} - \sigma_{k+d}\, v_\theta(z_{k+d},\, \sigma_{k+d},\, c), \tag{8}$$

$$\mathcal{L}_{\mathrm{CD}} = \mathrm{mean}_k\Big[\sqrt{\|\hat{x}^{\theta}_{0,k} - x_{0,k}\|^2 + c_0^2} - c_0\Big], \qquad c_0 = 0.00054\sqrt{D}, \tag{9}$$

where the target equation 7 carries a stop-gradient. The Pseudo-Huber form in equation 9 follows Song & Dhariwal (2024) and bounds the influence of the occasional very large residual that an ℓ_2 loss would allow to dominate an update. The student therefore reads a noisier state than the one its target is computed from, since $\sigma_{k+d} > \sigma_k$. The supervision band excludes the noisiest states, where $\hat{x}_0$ carries almost no structure, and the cleanest states, where the task is close to trivial. The student takes a single conditional forward pass, which absorbs classifier-free guidance into its weights and lets it be sampled at cfg 1, so the student is also a guidance-distilled model in the sense of Meng et al. (2023).

3.3 Decode-free projector reward

Selection determines which target the student is shown and leaves the student's own output unscored, so we add a reward term that compares it against the same reference.

The projector. Evaluating S on $\hat{x}^{\theta}_{0,k}$ directly would require a VAE decode and a DINOv2 forward pass at every training step. We therefore learn a projector $P_\phi : \mathbb{R}^{C\times H\times W} \to \mathcal{S}^{d-1}$ that maps a clean-latent estimate into the unit sphere of DINOv2 patch-mean space, reducing the reward to one small convolutional forward pass. P_ϕ is a stack of strided convolutional blocks with GroupNorm and SiLU, followed by spatial average pooling and an MLP head.

Objective. Let $\mathcal{K}_2$ denote the two supervised states of lowest noise level, that is the two smallest indices in $\mathcal{W}$. The training objective is

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{CD}} \;-\; \lambda \cdot \mathrm{mean}_{k \in \mathcal{K}_2} \big\langle P_\phi(\hat{x}^\theta_{0,k}),\; u(x^{\mathrm{ref}}) \big\rangle. \tag{10}$$

Restricting the reward to $\mathcal{K}_2$ matters, because at high noise $\hat{x}^\theta_0$ is too blurry for a DINO-space score to carry meaning and rewarding it injects noise into the update. We set λ so that the reward gradient is roughly 20% of the consistency gradient in norm, measured with a gradient probe over the first updates rather than tuned on the benchmark.

Pretraining the projector. We fit P_ϕ once, offline, on the teacher candidates of held-out captions that never enter the training pool, using an embedding term and a ranking term,

$$\mathcal{L}_P = \mathrm{mean}_j \big[1 - \cos(P_\phi(z_j),\, u(I_j))\big] \;+\; \mu\, \mathrm{KL}\Big(\mathrm{softmax}(\hat{S}/\tau) \,\Big\|\, \mathrm{softmax}(S/\tau)\Big), \tag{11}$$

where $\hat{S}_j = \langle P_\phi(z_j), u(x^{\mathrm{ref}}) \rangle$ and τ is the temperature at which candidates are compared. The two terms serve different purposes. The embedding term anchors the projected space to real DINO geometry and prevents P_ϕ from degenerating into an arbitrary map that happens to sort the training batch, while the ranking term targets the quantity the method actually consumes, which is an ordering rather than an absolute similarity. We select checkpoints by selection regret, defined as the RGB-score loss incurred by taking the projector's argmax instead of the true argmax, because regret reflects the downstream use more directly than embedding cosine does.

Refresh in alternation with the student. The projector is pretrained on teacher candidates, and the student's predictions leave that distribution as training proceeds, so a frozen projector gradually decalibrates. Every N updates we refit it for a small number of AdamW steps against the true RGB score of the student's most recent predictions, together with a replay batch of teacher candidates carrying the ranking term of equation 11 so that it retains the distribution it was pretrained on. This is the self-referential component identified in §2.2, since the reward is re-derived from the student's own outputs and scored by a frozen oracle.

The student and the projector are never updated in the same step, since P_ϕ is frozen during student updates and the student is frozen during a refresh. This prevents direct co-adaptation between the reward and the policy it scores and keeps the reward fixed within each student phase. No gradient flows from equation 10 into the DINO encoder or the reference embedding, both of which remain frozen throughout. In distributed training the refresh runs on the rank that holds the prediction buffer, and the updated projector is broadcast to the remaining ranks before the next student step, so every rank scores with identical parameters.

4 Experiments

We evaluate on three tasks. Compositional text-to-image generation on T2I-CompBench++ (§4.1) and on GenEval2 (§4.2) measures whether selection improves the text-image correspondence of a distilled student, and text-guided image editing (§4.3) measures whether that improvement transfers to a downstream use of the same model. Across all three the controlled comparison is against vanilla consistency distillation, an arm that shares every seed, the data order, the schedule, the supervision window and the gap distribution with ours. It differs in two respects, since it draws its trajectory uniformly among the candidates rather than by score and it trains without the reward term. Hyperparameters for our arms are given in Appendix B, and the offline and online costs of both arms in Appendix F.

4.1 Task 1: Compositional generation on T2I-CompBench++

T2I-CompBench (Huang et al., 2023) and its extension T2I-CompBench++ (Huang et al., 2025) score a model on prompts built to isolate compositional skills, grouped into attribute binding (colour, shape, texture), object relationships (two-dimensional and three-dimensional spatial, non-spatial), numeracy and complex compositions, each with its own evaluator. We report the official T2I-CompBench++ evaluators on all eight categories.

Two groups of baselines frame the result. Vanilla consistency distillation is the controlled comparison described above. The teacher is reported at eight steps, at its recommended setting of 40 steps, and at four steps without distillation, the last of which shows what the step budget costs. Table 1 reports all of them, with the per-category breakdown in Table 2. The ten-step teacher that produced the training targets is reported with the other teacher settings in Table 5.

Table 1: T2I-CompBench++ and GenEval2 results ($5k$ COCO captions, five seeds). Δ and p are prompt-paired against vanilla CD. Higher is better.

	Alignment				Fidelity (COCO val2017, 5k)			
Model	CompBench++	Δ (p)	GenEval2	Δ (p)	CMMD ↓	Prec. ↑	Rec. ↑	FID ↓
Vanilla CD, 4 steps (random trajectory, no reward)	0.4463	–	21.27	–	0.955	0.436	0.046	33.16
Ours, 4 steps (scored trajectory selection + reward)	**0.5036**	+0.0573 (<0.001)	**23.70**	+2.43 (0.02)	**0.701**	**0.481**	**0.122**	**26.80**
Teacher, 4 steps (no distillation)	0.2531	–	7.32	–	2.152	0.178	0.006	94.48
Teacher, 8 steps	0.4515	–	17.58	–	0.748	0.431	0.117	32.07
Teacher, 40 steps (recommended setting)	0.5027	–	18.51	–	0.633	0.560	0.161	27.08

Scored trajectory selection with the reward raises the T2I-CompBench++ mean from 0.4463 to 0.5036 (+0.0573, $p < 0.001$) and GenEval2 from 21.27 to 23.70 (+2.43, p=0.02), and it improves every fidelity metric at the same time, lowering FID from 33.16 to 26.80 and raising recall from 0.046 to 0.122. Alignment-directed fine-tuning often trades fidelity for alignment, as SoftREPA's FID rises from 31.6 to 36.2 while its alignment improves (Lee et al., 2025), and selecting among the teacher's own trajectories does not show that trade. The four-step student exceeds the eight-step teacher (0.4515) and the ten-step teacher that produced its training targets (0.4755), and it matches the teacher at its recommended 40-step setting (0.5027) with four network evaluations per image against 20 and 80. On GenEval2 every teacher setting in Table 1 scores below both students. These comparisons are to a single teacher sample. Selecting among the teacher's own samples at test time raises its score further, and selection at training and at test time compound (Table 5).

Per category (Table 2), the gain is largest on two-dimensional and three-dimensional spatial relations, followed by texture and numeracy, and it is smallest on non-spatial relations, where a photograph carries the least information about the prompt. Figure 2 shows examples on spatial-relation prompts.

Comparison with prior work. CTCal (Guo et al., 2026) is the closest published method on this benchmark, and Table 2 reproduces its T2I-CompBench++ results beside ours. CTCal fine-tunes SD3 (2B) and is evaluated at full sampler steps, whereas our student is SD3.5-Medium distilled to four steps and sampled at cfg 1.

Table 2: T2I-CompBench++ per category, with the published CTCal results (Guo et al., 2026).

	Color B-VQA	Shape B-VQA	Texture B-VQA	2D-Spatial UniDet	3D-Spatial UniDet	Numeracy UniDet	Non-Spatial Share-CoT	Complex 3-in-1
Base Model, 8 steps, w=7	0.7142	0.4988	0.6750	0.2425	0.3276	0.5518	0.7467	0.3163
Base Model, 40 steps, w=4.5	0.7953	0.5732	0.7244	0.2745	0.3549	0.5990	0.7787	0.3774
GORS (Huang et al., 2025)	0.8236	0.5833	0.7398	0.3232	0.4033	0.6280	0.7708	0.3739
CTCal (Guo et al., 2026)	0.8443	0.5968	0.7581	0.3476	**0.4117**	**0.6292**	0.7867	0.3814
Vanilla CD (4 steps)	0.7692	0.5158	0.6848	0.1432	0.2445	0.4924	0.7433	0.3561
Ours (4 steps)	**0.8730**	**0.6480**	**0.7901**	**0.3748**	0.3752	0.5979	**0.7980**	**0.3998**

4.2 Task 2: Compositional generation on GenEval2

GenEval2 (Ghosh et al., 2023; Kamath et al., 2025) decomposes each prompt into atomic requirements covering object presence, counting, attributes, position and verbs. Because a prompt scores near zero unless almost every atom is satisfied, the benchmark is more sensitive to complete compositional failure than T2I-CompBench is, and it reports difficulty as a function of the number of atoms. The baseline is again vanilla consistency distillation under the identical protocol, and we report the official evaluator.

The overall score in Table 4 rises from 21.3 to 23.7 (+2.4, p=0.02). The gain is carried by object presence (+8.1) and counting (+7.3), both at $p < 0.001$, while position (+1.0) and verbs (+0.8) do not change significantly. By prompt complexity, the score on three-atom prompts is unchanged (+1.1), and the gain is largest at four atoms (+8.9) and significant at every length from six to ten atoms.

4.3 Task 3: Text-guided image editing

Text-guided editing takes a source image with a source caption and a target caption that differs in a single concept, and asks the model to realise the target while preserving everything else. The task therefore scores two quantities at once, alignment to the target caption and fidelity to the source, and the two trade off as guidance strength rises. FlowEdit (Kulikov et al., 2025) performs this edit without inversion by constructing a direct path between the source and target distributions of a pretrained flow model, which makes it a natural harness for a distilled student.

SoftREPA (Lee et al., 2025) evaluates on PIE-Bench (Ju et al., 2024) and on 800 DIV2K images with LLaVA-generated source captions and Llama-edited target captions. Table 3 reproduces its FlowEdit results beside ours. PIE-Bench is evaluated with its official prompts and edit masks. The DIV2K prompts are rebuilt with SoftREPA's recipe, because the original prompts are not released. We use SoftREPA's metrics, namely ImageReward, PickScore, CLIP and HPS against the target caption and LPIPS against the source, and add the official PIE-Bench structure distance, PSNR and SSIM on PIE-Bench.

Table 3: Text-guided editing with FlowEdit (Kulikov et al., 2025). PSNR and SSIM are measured on the unedited region.

		Human Preference		Text Alignment			Structure	Background Preservation		
	Method	Image-Reward ↑	Pick-Score ↑	CLIP/ Edited ↑	CLIP/ Whole ↑	HPS ↑	Distance ↓	PSNR ↑	LPIPS/ Whole ↓	SSIM ↑
PIE-Bench	Vanilla CD (4-step student)	79.37	21.78	22.87	26.51	26.57	20.39	23.63	10.81	88.16
	FlowEdit	87.70	22.16	23.19	26.72	28.04	25.48	24.37	12.55	88.58
	FlowEdit + SoftREPA	102.24	**22.31**	23.60	27.19	**28.58**	24.07	24.99	12.60	**88.73**
	Ours (4-step student)	**102.68**	22.20	**23.81**	**27.45**	28.19	**15.79**	**25.23**	**11.23**	88.16
DIV2K	Vanilla CD (4-step student)	33.52	21.30	-	26.25	24.68	17.50	-	10.40	-
	FlowEdit	38.08	21.74	-	26.05	25.53	39.68	-	15.48	-
	FlowEdit + SoftREPA	46.68	**21.88**	-	26.39	26.03	35.66	-	14.98	-
	Ours (4-step student)	**66.90**	21.70	-	**27.26**	**26.68**	**19.35**	-	**13.26**	-

The student transfers to editing (Table 3). On PIE-Bench it improves on FlowEdit + SoftREPA in ImageReward, both CLIP scores, structure distance (15.79 against 24.07), PSNR and LPIPS, and trails in PickScore, HPS and SSIM. On DIV2K it improves in ImageReward (66.90 against 46.68), CLIP, HPS, structure distance (19.35 against 35.66) and LPIPS, and trails only in PickScore (21.70 against 21.88). Against the vanilla consistency-distilled student under the same editing procedure, ours raises ImageReward from 79.37 to 102.68 on PIE-Bench and from 33.52 to 66.90 on DIV2K. The vanilla student has lower LPIPS on both datasets and lower structure distance on DIV2K, while ours has lower structure distance and higher PSNR on PIE-Bench. Figure 3 shows examples, and Figure 5 in the appendix gives further ones.

5 Ablations and Diagnostics

Trajectory scoring and the reward. A 2×2 factorial on the COCO 3k configuration separates the two components of the method, with five seeds per arm (Appendix E.4, Table 7). Each component helps on its own. Trajectory scoring raises the T2I-CompBench++ mean by 0.0117 and the reward by 0.0068, and together they improve on vanilla distillation by 0.0172. A two-way analysis over the 20 runs finds significant main effects for scoring ($p = 5 \times 10^{-6}$) and for the reward ($p = 0.002$), so the reward adds to what selection provides rather than duplicating it.

Reward variants. Table 6 in Appendix E isolates the reward design. Every variant beats vanilla CD, the differences between them are small relative to seed noise, and only the refreshed-16-step

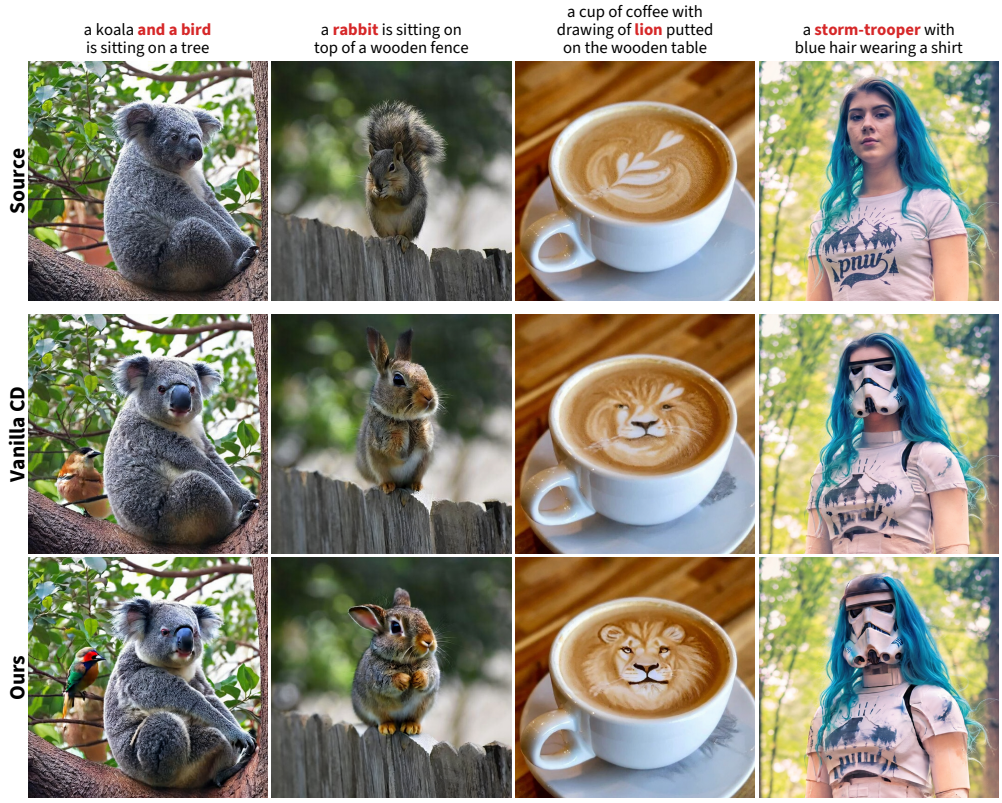

Figure 3: **Text-guided editing with FlowEdit** (Kulikov et al., 2025), using the four-step students as backbones. Each column gives a source image and a target caption, with the edited concept in red. Rows show the source, the vanilla consistency-distilled student, and ours. Quantitative results are in Table 3.

variant reaches $p < 0.01$. A frozen projector already recovers $+0.0109$, so most of the benefit comes from selection together with a reward rather than from the refresh itself, and refreshing mainly reduces variance (sd $0.0075 \rightarrow 0.0017$).

Scaling to 118k captions. We repeat the comparison with the 118k-caption pool (three seeds). The gains hold at this scale: CompBench improves by $+0.0352 \pm 0.0044$ (p=0.005, seed-paired) and GenEval2 by $+3.9 \pm 0.6$ (p=0.007), with better fidelity (CMMD -0.15, precision $+0.07$, recall $+0.03$). Appendix G gives the per-category, GenEval2 and fidelity breakdowns (Tables 11, 9 and 10).

6 Conclusion

Few-step distillation has an unexamined degree of freedom: *which* of the teacher's possible trajectories the student imitates. Choosing that trajectory by a text-free DINO score against the caption's own photograph, and rewarding the student's clean-latent estimates through a decode-free projector, improves compositional alignment on T2I-CompBench++ ($0.4463 \rightarrow 0.5036$) and GenEval2 ($21.3 \rightarrow 23.7$) without a fidelity cost, and lets a four-step student exceed the ten-step teacher it was distilled from and match the 40-step teacher at its recommended setting. We discuss the limitations of the method in Appendix D.

AI USAGE CLARIFICATION

We used large language models (LLMs) as a general-purpose writing and coding assistant. Specifically, LLMs were used to help organise the sectioning and structure of the manuscript, to improve clarity and readability, and to assist with routine software engineering tasks such as writing boilerplate code, refactoring, and debugging. The problem formulation, scored trajectory selection, the decode-free projector reward and its refresh schedule, and the experiments originate from the authors. Every LLM-assisted component was reviewed, verified, and edited by the authors. The authors take full responsibility for all content of this submission.

ETHICS STATEMENT

All experiments use publicly available data, namely COCO train2017 captions and photographs for training and COCO val2017 for fidelity (Lin et al., 2014), and the T2I-CompBench++ (Huang et al., 2025), GenEval2 (Kamath et al., 2025), PIE-Bench (Ju et al., 2024) and DIV2K benchmarks for evaluation. COCO photographs include people, and we use them only as unmodified references for a feature-space score. Our method distils an existing text-to-image model, SD3.5-Medium (Esser et al., 2024), rather than training a new one. Faster sampling lowers the cost of producing images, including misleading ones, and the student inherits the biases of its teacher and of the COCO reference set.

REPRODUCIBILITY STATEMENT

We provide our implementation, including candidate-cache construction, projector pretraining, distillation and evaluation scripts, and the scripts that reproduce the tables and figures in this paper, as supplementary material. The training procedure is given in Algorithm 1 (Appendix C), and all hyperparameters, model versions and evaluation settings are listed in Appendix B, with the training and inference cost of each arm in Appendix F. Baseline results for GORS, CTCal, FlowEdit and SoftREPA are reported from the original papers (§4.1 and §4.3), and all other numbers are produced by us.

REFERENCES

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *International Conference on Machine Learning (ICML)*, 2025.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Ramón Calvo-González and François Fleuret. Laminating representation autoencoders for efficient diffusion. arXiv:2602.04873, 2026.

Shipeng Cao, Biao Qian, Haipeng Liu, Yang Wang, and Meng Wang. AI-T2I: Aggregating-and-isolating cross-attention to diffusion models for text-to-image synthesis. *IEEE TMM*, 2026. doi: 10.1109/TMM.2026.3717482. URL https://doi.org/10.1109/TMM.2026.3717482.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.html.

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 TOG*, 42(4), 2023. doi: 10.1145/3592116. URL https://doi.org/10.1145/3592116.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.html.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, pp. 12606–12633, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. InitNO: Boosting text-to-image diffusion models via initial noise optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xiefan Guo, Xinzhu Ma, Haiyu Zhang, and Di Huang. CTCal: Rethinking text-to-image diffusion models via cross-timestep self-calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026, Denver, CO, USA, June 3-7, 2026*, pp. 43558–43567. Computer Vision Foundation, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_CTCal_Rethinking_Text-to-Image_Diffusion_Models_via_Cross-Timestep_Self-Calibration_CVPR_2026_paper.html.

Yasaman Haghighi, Bastien van Delft, Mariam Hassan, and Alexandre Alahi. LayerSync: Self-aligning intermediate layers. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/32eb183794ef5ef9a3ab1d40a3d2b303-Abstract-Conference.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f8ad010cdd9143dbb0e9308c093aff24-Abstract-Datasets_and_Benchmarks.html.

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 Trans. Pattern Anal. Mach. Intell.*, 47(5):3563–3579, 2025. doi: 10.1109/TPAMI.2025.3531907. URL https://doi.org/10.1109/TPAMI.2025.3531907.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven C. H. Hoi. D-OPSD: On-policy self-distillation for continuously tuning step-distilled diffusion models. arXiv preprint arXiv:2605.05204, 2026. URL https://arxiv.org/abs/2605.05204.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=FoMZ4ljhVw.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang 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 *ICLR*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c204d12afa0175285e5aac65188808b4-Abstract-Conference.html.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, pp. 19721–19730, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kulikov_FlowEdit_Inversion-Free_Text-Based_Editing_Using_Pre-Trained_Flow_Models_ICCV_2025_paper.html.

Jaa-Yeon Lee, Byunghee Cha, Jeongsol Kim, and Jong Chul Ye. Aligning text to image in diffusion models is easier than you think. In *NeurIPS*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e67931070e119ef84f40048f1b19d39e-Abstract-Conference.html.

Jaa-Yeon Lee, Yeobin Hong, Taesung Kwon, and Jong Chul Ye. Alignment-guided score matching for text-to-image alignment in diffusion models. In *ICML*, 2026. URL https://arxiv.org/abs/2605.30038. Spotlight.

Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. Enhancing compositional text-to-image generation with reliable random seeds. In *ICLR*, 2025. URL https://openreview.net/forum?id=5BSlakturs.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, pp. 366–384, 2024. doi: 10.1007/978-3-031-72673-6_20. URL https://doi.org/10.1007/978-3-031-72673-6_20.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

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. URL https://arxiv.org/abs/2310.04378.

Weijian Luo, Zemin Huang, Zhengyang Geng, J. Zico Kolter, and Guo-Jun Qi. One-step diffusion distillation through score implicit matching. In *NeurIPS*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d107ca794d83c8242e357e6a43a068f4-Abstract-Conference.html.

Zhengyao Lv, Tianlin Pan, Chenyang Si, Zhaoxi Chen, Wangmeng Zuo, Ziwei Liu, and Kwan-Yee K. Wong. Rethinking cross-modal interaction in multimodal diffusion transformers. In *ICCV*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lv_Rethinking_Cross-Modal_Interaction_in_Multimodal_Diffusion_Transformers_ICCV_2025_paper.html.

Zhiyuan Ma, Yuzhu Zhang, Guoli Jia, Liangliang Zhao, Yichao Ma, Mingjie Ma, Gaofeng Liu, Kaiyan Zhang, Ning Ding, Jianjun Li, and Bowen Zhou. Efficient diffusion models: A comprehensive survey from principles to practices. *IEEE TPAMI*, 47(9):7506–7525, 2025. doi: 10.1109/TPAMI.2025.3569700. URL https://ieeexplore.ieee.org/document/11002717.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Meng_On_Distillation_of_Guided_Diffusion_Models_CVPR_2023_paper.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

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 *ICLR*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. arXiv preprint arXiv:2310.03739, 2023. URL https://arxiv.org/abs/2310.03739.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, 2024. URL https://doi.org/10.1007/978-3-031-73016-0_6.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3929a7785bd56f57edcff0152ab41289-Abstract-Conference.html.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *ICLR*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html.

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 *CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wallace_Diffusion_Model_Alignment_Using_Direct_Preference_Optimization_CVPR_2024_paper.html.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, Kai Wang, and Yang You. REPA works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. In *NeurIPS*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c8033b6ed6dad2af747fc8fb48525996-Abstract-Conference.html.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *NeurIPS*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0b99315234cc95e6ef281f9155b68832-Abstract-Conference.html. Oral.

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 *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

Yuan Zhang, Chenyi Li, Haodong Yu, Guoqing Ma, Jiajun Zha, Yuanming Yang, Bo Wang, Wei Tang, Wenbo Li, Haoyang Huang, and Nan Duan. Teacher-feature drifting: One-step diffusion distillation with pretrained diffusion representations. arXiv preprint arXiv:2605.07327, 2026. URL https://arxiv.org/abs/2605.07327.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024. URL https://proceedings.mlr.press/v235/zhou24x.html.

Qin Zhou, Zhiyang Zhang, Jinglong Wang, Xiaobin Li, Jing Zhang, Qian Yu, Lu Sheng, and Dong Xu. ELBO-T2IAlign: A generic ELBO-based method for calibrating pixel-level text-image alignment in diffusion models. *IEEE TIP*, 35:7319–7331, 2026. doi: 10.1109/TIP.2026.3709461. URL https://doi.org/10.1109/TIP.2026.3709461.

A Related Work

Selecting supervision rather than reproducing it. The distillation methods surveyed in §2.1 take the teacher's output as the target and use a single sampled trajectory without regard to its utility. Selection does appear elsewhere in the pipeline. GORS (Huang et al., 2023) and CTCal (Guo et al., 2026) both fine-tune on reward-filtered self-generated images, and ranking a small candidate pool at inference time is a strong and widely used baseline (Lin et al., 2024). Our contribution moves that selection inside the distillation loop, where it is close to free because the candidates are generated anyway, and applies it to trajectories rather than to final images.

Representation alignment. REPA aligns a diffusion transformer's hidden states to DINOv2 features and sharply accelerates training (Yu et al., 2025). Several extensions follow this template: REG entangles a foundation-model class token with the image latents (Wu et al., 2025), LayerSync aligns intermediate diffusion layers to one another and decreases the external encoder (Haghighi et al., 2026), and Wang et al. (2025) show the benefit plateaus late in training. SoftREPA reframes the objective contrastively with a small set of soft text tokens and reports gains on both generation and editing (Lee et al., 2025), which makes it our editing baseline in §4.3. All of these supply a per-sample alignment target for the representation. We take the same class of encoder and reduce it to a scalar that ranks trajectories and scores the student's estimates, so the encoder shapes the training signal without ever defining a per-token target.

Text-image alignment fine-tuning. A growing body of work targets text-image alignment directly. Inference-time methods steer the attention maps of a frozen model, as in Attend-and-Excite (Chefer et al., 2023), and training-time methods change what the model learns. CTCal (Guo et al., 2026), our reference point on T2I-CompBench++, calibrates learning at large timesteps using the more reliable cross-attention maps formed at small ones. AI-T2I (Cao et al., 2026) consolidates scattered intra-token activations and separates inter-token ones so that each subject occupies a coherent region. TACA (Lv et al., 2025) rebalances cross-modal attention inside multimodal diffusion transformers, and ELBO-T2IAlign (Zhou et al., 2026) calibrates pixel-level correspondence through an ELBO-derived likelihood. Alignment-guided score matching (Lee et al., 2026) folds contrastive guidance into the score-matching objective and thereby decreases the instability that arises when negative pairs are penalised too strongly. All of these read or reshape the model's text-conditioned internals and therefore require a text encoder at training time. Our signal is one text-free scalar per candidate, computed from a photograph.

Reward fine-tuning. DDPO (Black et al., 2024) and AlignProp (Prabhudesai et al., 2023) optimise an RL-estimated or differentiable reward, and Diffusion-DPO (Wallace et al., 2024) learns from pairwise human preference. Each of these depends on a trained reward model (Xu et al., 2023) or on preference annotation. Our reward is a cosine similarity to a reference photograph that any image-caption corpus already provides, and it enters as one term inside a distillation loss that remains the primary objective.

B Implementation Details

Models and sampling. The base model is SD3.5-Medium. The teacher runs $K{=}10$ Euler steps at classifier-free guidance $w{=}7$ and generates $J{=}4$ candidates per caption, seeded by $\mathrm{idx} \cdot 1000 + j$. Students are distilled to four steps and sampled at cfg 1, since a single conditional forward pass absorbs guidance into the student's weights.

Distillation. Supervision covers teacher states in the window $[0.4K, 0.9K]$ with step gap $d \sim \mathcal{U}\{1, 2, 3\}$. The Pseudo-Huber constant is $c_0 = 0.00054\sqrt{D}$ with D the latent dimension. Training captions are COCO train2017 (Lin et al., 2014) paired with their photographs.

Scorer. The reference score uses DINOv2 patch tokens with the CLS token dropped, mean-pooled and ℓ_2-normalised. Candidate scores are computed once offline and cached as scalars.



Figure 4: **Additional Spatial-relation prompts from T2I-CompBench++**, with the relation word in red. Rows show our four-step student, the vanilla consistency-distilled student at four steps, and the SD3.5-Medium base model. The examples illustrate the spatial-relation gains quantified in Table 2.



Figure 5: **Further text-guided editing examples** with FlowEdit, in the layout of Figure 3: source, vanilla consistency-distilled student, and ours.

Projector. The projector maps a $16 \times 64 \times 64$ latent to a unit vector in $\mathbb{R}^{768}$ through strided convolutional blocks with GroupNorm and SiLU, followed by spatial average pooling and a two-layer MLP head, for a total of 10.7M parameters. It is pretrained for 30 epochs with AdamW at learning rate 3×10^{-4}, weight decay 0.01 and a one-cycle schedule, on the candidates of 22k held-

out captions that never enter the training pool. The ranking temperature is $\tau = 0.04$ and the ranking term carries weight $\mu = 1$. Checkpoints are selected by lowest selection regret on a held-out split.

Reward and refresh. The reward acts on the two least-noisy supervised states with weight $\lambda = 80$, chosen so that the reward gradient is roughly 20% of the consistency gradient in norm, as measured by a gradient probe over the first updates. The projector is refreshed every N=100 optimizer updates for M=16 AdamW steps at learning rate 10^{-4}, against the decoded RGB DINOv2 score of the 32 most recent student predictions plus a replay batch of 8 cached teacher candidates carrying the ranking term of equation 11. The refresh runs on the rank holding the prediction buffer and the updated projector is broadcast to the remaining ranks before the next student update.

Evaluation. The 3k ablations use three seeds (five in the 2×2 ablation) and the 118k study three seeds, with seed-paired means, standard deviations and two-sided t-tests. Fidelity is measured against 5,000 COCO val2017 captions.

C Training Algorithm

Algorithm 1 summarises one training step of scored consistency distillation, including the offline construction of the candidate cache and the periodic projector refresh described in §3.3.

Algorithm 1 Scored consistency distillation with a refreshed projector reward

Require: frozen teacher $v_{\mathcal{T}}$, student v_θ, projector P_ϕ, frozen DINO encoder u, captions with photographs $\{(\mathrm{idx}, c, x^{\mathrm{ref}})\}$

Require: candidates J, teacher steps K, guidance w, reward weight λ, refresh period N, refresh steps M

Offline, once. For each caption, roll J teacher trajectories from seeds $\mathrm{idx} \cdot 1000 + j$, decode, and cache $S_j = \cos(u(I_j), u(x^{\mathrm{ref}}))$.

Offline, once. Pretrain P_ϕ on held-out captions with equation 11 and keep the checkpoint of lowest selection regret.

for each training step t **do**

  sample a caption; read cached scores and set $j^\star \leftarrow \arg\max_j S_j$

  regenerate trajectory $\{z_k\}$ of candidate $j^\star$ from its seed ▷ no grad

  $x_{0,k} \leftarrow z_k - \sigma_k (z_k - z_{k-1})/(\sigma_k - \sigma_{k-1})$ for $k \in \mathcal{W}$ ▷ targets, cleaner states

  sample $d \sim \mathcal{U}\{1, 2, 3\}$ and compute $\hat{x}^\theta_{0,k}$ from the noisier state z_{k+d} by equation 8

  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{CD}}$ by equation 9

  $\mathcal{L} \leftarrow \mathcal{L} - \lambda \cdot \mathrm{mean}_{k \in \mathcal{K}_2} \langle P_\phi(\hat{x}^\theta_{0,k}), u(x^{\mathrm{ref}}) \rangle$ ▷ P_ϕ frozen

  update θ on $\mathcal{L}$; append recent $\hat{x}^\theta_0$ to buffer $\mathcal{B}$

  if $t \bmod N = 0$ **then**

    decode $\mathcal{B}$ and score with the frozen DINO encoder ▷ θ frozen

    refit P_ϕ for M steps on $\mathcal{B}$ plus a replay batch of teacher candidates

    broadcast P_ϕ to all ranks

  end if

end for

D Limitations

The method needs an image-caption corpus with real photographs, so it does not apply directly to caption-only data. For such data, an open- or closed-source text-to-image model can serve as an oracle that generates the reference images. The score carries little information about relations that are not visible in appearance, which is why the non-spatial category gains least. Building the candidate pool costs J teacher rollouts per caption, so the offline budget grows linearly in J (Appendix F). All results use a single base model distilled to four steps.

E Additional Results and Ablation

E.1 Main Results Breakdown Per Category

Table 4: GenEval2 breakdown for the models of Table 1 (800 prompts, one image per prompt, prompt-paired p-values). Overall is the official per-prompt geometric mean.

GenEval2 measure ($\times 100$)	Vanilla CD	Ours	Ours − vanilla (p)
Overall Average	21.3	23.7	$+\mathbf{2.4}$ (0.02)
Skill: object	78.5	86.6	$+\mathbf{8.1}$ (<0.001)
Skill: count	34.4	41.6	$+\mathbf{7.3}$ (<0.001)
Skill: position	49.4	50.4	$+1.0$ (0.53)
Skill: verb	24.6	25.4	$+0.8$ (0.84)
Atoms $= 3$	70.5	71.6	$+1.1$ (0.58)
Atoms $= 4$	68.1	77.0	$+\mathbf{8.9}$ (<0.001)
Atoms $= 5$	65.7	68.6	$+\mathbf{2.9}$ (0.07)
Atoms $= 6$	62.5	69.6	$+\mathbf{7.1}$ (<0.001)
Atoms $= 7$	56.9	63.1	$+\mathbf{6.2}$ (<0.001)
Atoms $= 8$	56.1	61.7	$+\mathbf{5.6}$ (<0.001)
Atoms $= 9$	53.1	58.5	$+\mathbf{5.4}$ (<0.001)
Atoms $= 10$	49.1	52.2	$+\mathbf{3.1}$ (0.02)

E.2 Ablation: test-time selection

Table 5 places the comparison against test-time selection. With the official evaluator choosing among the teacher's own samples, the ten-step teacher reaches 0.5931 at best-of-4 and 0.6428 at best-of-10, and a single sample of our student (0.5036) recovers 24% of the best-of-4 gain over the teacher's single sample. Selection at training time and at test time compound: at best-of-10 our student (0.6482) exceeds the ten-step teacher (0.6428) and approaches the 28- and 40-step teachers (0.6528, 0.6529).

Table 5: Single-sample and oracle best-of-J T2I-CompBench. For best-of-J the category's official evaluator picks the best of J candidates; for $J{=}4$ the value is the exact expectation over all 4-subsets of the ten.

Model	1 sample	Best of 4 (oracle)	Best of 10 (oracle)
Vanilla CD (4 steps)	0.4463	0.5592	0.6113
Ours (4 steps)	0.5036	0.6052	0.6482
Teacher, 8 steps, $w{=}7$	0.4515	0.5767	0.6278
Teacher, 10 steps, $w{=}7$ (the distillation source)	0.4755	0.5931	0.6428
Teacher, 28 steps, $w{=}7$	0.5110	0.6100	0.6528
Teacher, 40 steps, $w{=}4.5$ (recommended setting)	0.5027	0.6079	0.6529

E.3 Ablation: reward variants

Table 6: Reward-variant ablation on the COCO-caption configuration ($3k$ captions, three seeds; the reward's target is the caption's photograph).

Reward variant	CompBench	Δ vs. vanilla (p)	GenEval2	Δ vs. vanilla (p)	CMMD↓
Vanilla CD (no selection, no reward)	0.4738 ± 0.0017	–	22.5 ± 0.9	–	0.837
Projector frozen	0.4847 ± 0.0075	$+0.0109 \pm 0.0075$ (0.13)	24.2 ± 0.4	$+1.66 \pm 0.60$ (0.04)	0.810
Refresh every 100 updates, 4 steps	0.4876 ± 0.0096	$+0.0138 \pm 0.0093$ (0.12)	24.0 ± 0.3	$+1.44 \pm 1.02$ (0.14)	0.843
Refresh every 25 updates, 4 steps	0.4866 ± 0.0029	$+0.0128 \pm 0.0037$ (0.03)	23.4 ± 0.3	$+0.85 \pm 0.71$ (0.17)	0.833
Refresh every 100 updates, 16 steps	0.4887 ± 0.0025	$+0.0149 \pm 0.0017$ (0.004)	23.5 ± 1.6	$+1.00 \pm 0.91$ (0.20)	0.817

E.4 Ablation: trajectory scoring and reward

We isolate the two components of our method with a 2×2 factorial on the COCO 3k configuration. Teacher trajectories are either picked at random, as in vanilla consistency distillation, or selected by the DINOv2 mean patch score. Each variant is trained with and without the DINOv2 reward on the student's clean estimates. All four arms share the captions, candidate cache, training schedule (6,000 updates) and seeds. We train five seeds per arm, and evaluate the 4-step students on T2I-CompBench++.

Table 7 shows that each component helps on its own and that their gains add up. Trajectory scoring raises the CompBench mean by 0.0117 and the reward by 0.0068; together they improve on naive distillation by 0.0172. A two-way analysis over the 20 runs finds significant main effects for scoring ($+0.011$, $p = 5 \times 10^{-6}$) and for the reward ($+0.006$, $p = 0.002$). The reward therefore adds to what trajectory selection already provides rather than duplicating it.

Table 7: Ablation of trajectory scoring and the reward (COCO 3k configuration, 4 steps). T2I-CompBench++ mean $\pm$ std over five training seeds; Δ is relative to vanilla consistency distillation.

DINO scoring	Reward	CompBench ↑	Δ
		0.4751 ± 0.0032	–
	✓	0.4819 ± 0.0060	$+0.0068$
✓		0.4868 ± 0.0031	$+0.0117$
✓	✓	$\mathbf{0.4923 \pm 0.0021}$	$\mathbf{+0.0172}$

F Cost and Data

Table 8 separates three budgets, namely offline cache construction, projector construction, and online student training, and reports inference cost separately. Building the candidate pool costs J teacher rollouts per caption, so the offline budget grows linearly in J even though the online student schedule is unchanged. The projector is pretrained on 22k auxiliary captions that never enter the student optimizer.

G Results at 118k Captions

G.1 GenEval2 at 118k captions

Table 9: GenEval2 breakdown at the 118k-caption scale (three seeds, seed-paired t-tests).

GenEval2 measure ($\times 100$), 118k, 3 seeds	Vanilla CD	Ours	Ours − vanilla (p)
Overall (official; per-prompt geometric mean)	19.1 ± 0.6	23.0 ± 1.1	$+3.9 \pm 0.6$ (0.007)
Skill: object (per atom)	79.5 ± 1.2	85.9 ± 0.6	$+6.3 \pm 1.6$ (0.02)
Skill: count (per atom)	36.5 ± 1.2	39.8 ± 0.9	$+3.3 \pm 0.6$ (0.01)
Skill: attribute (per atom)	68.0 ± 1.2	71.8 ± 0.3	$+3.8 \pm 1.5$ (0.05)
Skill: position (per atom)	43.3 ± 1.1	48.7 ± 1.3	$+5.4 \pm 2.4$ (0.06)
Skill: verb (per atom)	18.2 ± 0.8	25.1 ± 1.3	$+6.9 \pm 1.6$ (0.02)
Atoms = 3 (per atom)	69.9 ± 1.9	72.1 ± 2.3	$+2.2 \pm 1.9$ (0.19)
Atoms = 4 (per atom)	70.4 ± 1.1	75.5 ± 0.9	$+5.1 \pm 1.0$ (0.01)
Atoms = 5 (per atom)	64.9 ± 1.6	69.1 ± 0.6	$+4.1 \pm 2.0$ (0.07)
Atoms = 6 (per atom)	62.4 ± 0.5	68.3 ± 0.5	$+5.9 \pm 0.1$ (<0.001)
Atoms = 7 (per atom)	58.7 ± 1.0	62.3 ± 1.9	$+3.6 \pm 1.8$ (0.08)
Atoms = 8 (per atom)	54.7 ± 1.5	59.2 ± 1.5	$+4.6 \pm 1.0$ (0.02)
Atoms = 9 (per atom)	52.3 ± 0.5	58.6 ± 0.6	$+6.4 \pm 0.2$ (<0.001)
Atoms = 10 (per atom)	46.9 ± 0.8	51.2 ± 0.3	$+4.3 \pm 0.9$ (0.02)

Table 8: Training and inference cost by budget. J is the number of teacher candidates per caption and K the number of teacher sampler steps. Inference cost is identical across arms.

	Vanilla CD	Selection only	Reward only	Full method
(a) Offline cache construction, once per caption pool				
Teacher NFEs per caption	$2K$	$2JK$	$2K$	$2JK$
VAE decodes per caption	0	J	0	J
DINO calls per caption	0	$J{+}1$	1	$J{+}1$
GPU-hours	0.30	0.89	0.30	0.89
Cache size	0.7 MB	1.1 MB	1.1 GB	1.1 GB
(b) Projector construction, once				
Auxiliary captions used	0	0	22k	22k
Teacher NFEs	0	0	1.4M	1.4M
GPU-hours	0	0	5.4	5.4
(c) Online student training				
Student NFEs per update	64	64	64	64
Teacher NFEs per update	320	320	320	320
Projector / decode calls	0	0	32 / 0.3	32 / 0.3
Refresh overhead (% of step)	0	0	3.5	3.5
Peak memory	66 GB	66 GB	75 GB	75 GB
Wall-clock per 1k updates	51 min	51 min	53 min	53 min
(d) Inference				
Sampler steps	4	4	4	4
Guidance passes per step	1	1	1	1
Total GPU-hours	0.7	0.8	1.8	1.9

G.2 FIDELITY AT 118K CAPTIONS

Table 10: Fidelity at the 118k-caption scale against 5,000 COCO val2017 captions (three seeds, seed-paired t-tests)

118k, three seeds	CMMD ↓	Precision ↑	Recall ↑	FID ↓
Vanilla CD	0.84/0.84/0.83	0.485/0.489/0.503	0.059/0.066/0.062	31.08/30.97/31.44
Ours	0.69/0.68/0.68	0.559/0.572/0.558	0.085/0.091/0.086	30.45/30.66/31.25
Ours − vanilla (p)	-0.153 ± 0.006 (<0.001)	$+0.071 \pm 0.014$ (0.01)	$+0.025 \pm 0.001$ (<0.001)	-0.38 ± 0.23 (0.10)

G.3 PER-CATEGORY COMPBENCH AT 118K CAPTIONS

Table 11: Per-category T2I-CompBench at the 118k-caption scale (one image per prompt, three seeds, seed-paired t-tests.)

Category	Vanilla CD	Ours	Ours − vanilla (p)
Color	0.7633 ± 0.0117	0.8101 ± 0.0074	$+0.0468 \pm 0.0088$ (0.01)
Shape	0.5361 ± 0.0067	0.5459 ± 0.0153	$+0.0097 \pm 0.0204$ (0.50)
Texture	0.6874 ± 0.0114	0.7379 ± 0.0038	$+0.0505 \pm 0.0109$ (0.02)
2D-spatial	0.1867 ± 0.0110	0.2362 ± 0.0011	$+0.0495 \pm 0.0103$ (0.01)
3D-spatial	0.2861 ± 0.0062	0.3342 ± 0.0048	$+0.0480 \pm 0.0109$ (0.02)
Numeracy	0.5358 ± 0.0107	0.5885 ± 0.0098	$+0.0528 \pm 0.0202$ (0.05)
Non-spatial	0.3126 ± 0.0008	0.3146 ± 0.0007	$+0.0019 \pm 0.0014$ (0.15)
Complex	0.3543 ± 0.0017	0.3770 ± 0.0009	$+0.0227 \pm 0.0012$ (<0.001)
Mean	0.4578 ± 0.0011	0.4930 ± 0.0033	$+0.0352 \pm 0.0044$ (0.005)

H SELECTION-RULE VARIANTS

Best-of-J and exponential tilting. Best-of-J is related to, but distinct from, the exponentially tilted distribution

$$q_\beta(\tau \mid c, x^{\mathrm{ref}}) \;\propto\; p_{\mathcal{T}}(\tau \mid c)\, \exp\big(\beta\, r(\tau)\big), \tag{12}$$

which weights trajectories by the value of their score rather than its rank. The two are connected through the softmax rule over a sampled candidate set, which weights candidate j by $\exp(\beta r_j)/\sum_k \exp(\beta r_k)$. At fixed β this rule is a self-normalised importance-sampling estimate of q_β and converges to it as $J \to \infty$. At fixed J its limit as $\beta \to \infty$ is the argmax over the candidates, which is best-of-J. The two limits do not commute, and at finite J best-of-J is not in general equal to q_β for any β. We use the argmax, and we treat equation 12 as the separately defined target of the finite-temperature softmax variants reported in Table 12.

Divergence of the best-of-J distribution. Under $p_{\mathrm{best},J}$ in equation 5, the quantity $F_r(r(\tau))$ is distributed as the maximum of J independent uniform variables, so $\mathbb{E}_{p_{\mathrm{best},J}}[\log F_r] = -1/J$. Substituting into the definition of the divergence gives

$$\mathrm{KL}\big(p_{\mathrm{best},J} \,\big\|\, p_{\mathcal{T}}\big) = \log J + (J-1)\,\mathbb{E}_{p_{\mathrm{best},J}}\big[\log F_r\big] = \log J - \frac{J-1}{J} \tag{13}$$

for continuous scores. When scores can tie the expression becomes an upper bound on the actual divergence (Beirami et al., 2025). For $J = 4$ it equals $\log 4 - 3/4 \approx 0.64$ nats, and it grows only logarithmically, reaching ≈ 1.20 nats at $J = 8$ and ≈ 1.83 at $J = 16$. The bound concerns the supervision distribution and not the student's learned sampling distribution.

Selection rules. Beyond argmax selection we implemented a family of Boltzmann rules that weight or sample candidates by $\mathrm{softmax}(S/T)$ rather than taking the maximum; Table 12 reports all of them on the 3k configuration. Exact Boltzmann weighting (all four candidates, each loss weighted) shows no improvement over argmax at $T \in \{0.04, 0.08\}$ (-0.0033 and -0.0107, neither significant) at three times the training compute, and weighting all four candidates uniformly recovers none of the gain ($+0.0004$ over random). Sampling one candidate per visit, or freezing one Boltzmann draw per caption, lands 0.0014 to 0.0067 below argmax. A decode-free *latent* selector using the projector of §3.3 recovers roughly three quarters of the argmax gain ($+0.0103$ against $+0.0136$ over random), which bounds how much of the effect survives without decoding candidates at cache-build time.

Table 12: Selection rules on the COCO $3k$ configuration. Δ is seed-paired with a two-sided t-test; Train GPU-h is the training cost of one run.

Selection rule	CompBench	Δ vs. random (p)	Δ vs. argmax (p)	GenEval2
Random draw (vanilla CD)	0.4737 ± 0.0017	–	–	22.5 ± 1.0
Argmax (the rule used in ours)	0.4873 ± 0.0032	$+0.0136$ (0.03)	–	23.7 ± 1.6
Boltzmann, exact weighting, T=0.04	0.4840 ± 0.0030	$+0.0102$ (0.03)	-0.0033 (0.45)	23.7 ± 1.0
Boltzmann, exact weighting, T=0.08	0.4766 ± 0.0089	$+0.0029$ (0.65)	-0.0107 (0.24)	23.5 ± 0.7
Uniform, exact weighting (all J)	0.4741 ± 0.0054	$+0.0004$ (0.93)	-0.0132 (0.08)	23.5 ± 0.7
Boltzmann, sampled per visit, T=0.04	0.4831 ± 0.0043	$+0.0094$ (0.07)	-0.0042 (0.07)	23.9 ± 0.4
Boltzmann, sampled per visit, T=0.08	0.4806 ± 0.0020	$+0.0069$ (0.08)	-0.0067 (0.07)	23.5 ± 1.1
Uniform, sampled per visit	0.4749 ± 0.0029	$+0.0012$ (0.59)	-0.0124 (0.006)	22.2 ± 0.8
Boltzmann, frozen draw, T=0.04	0.4859 ± 0.0031	$+0.0121$ (0.03)	-0.0014 (0.07)	24.2 ± 0.6
Latent (projector) score, argmax	0.4840 ± 0.0058	$+0.0103$ (0.14)	-0.0033 (0.35)	22.8 ± 0.3
Latent score, exact weighting, T=0.04	0.4808 ± 0.0072	$+0.0070$ (0.25)	-0.0065 (0.38)	23.3 ± 0.8