REVISITING MOMENTUM TEACHERS FOR PIXEL-SPACE GENERATIVE SELF-ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative self-alignment uses a generator's own features as training targets, typically obtained from an exponential-moving-average (EMA) teacher. We revisit whether this momentum teacher is necessary for pixel-space generation and introduce SiamFlow, which uses the current generator to process both a noisier student view and a cleaner target view. A shared projector and a student-only predictor align shallow student features with deeper, stop-gradient targets. On ImageNet 256×256 with JiT-L/16, SiamFlow achieves FID 2.13, compared with 2.33 when only the teacher backbone uses EMA weights and 2.39 without alignment. It approaches PixelREPA's reported 2.11 without an external semantic encoder and improves JiT at larger model size and image resolution. Controlled ablations identify how variance control, feature rescaling, and alignment depth support online self-alignment. Variance regularization improves an unstable online configuration, while faster EMA updates reduce target differences without closing the FID gap. Higher cross-noise retrieval accuracy also does not consistently translate into lower generation FID. These results show that effective pixel-space self-alignment need not rely on a momentum teacher or change the sampling architecture.



Figure 1: **SiamFlow: online self-alignment with competitive generation quality.** Left: SiamFlow shares one online backbone; EMA alignment maintains a separate teacher. Right: best FID of JiT-L/16 recipes on ImageNet 256^2, 600 epochs. SRA and Self-Flow use EMA; "no mask" removes the Self-Flow token mask. SiamFlow approaches PixelREPA's 2.11 without an external semantic encoder.

1 INTRODUCTION

Representation alignment improves diffusion transformers by supervising their intermediate features with a pretrained encoder (Yu et al., 2025a; Peebles & Xie, 2023; Ma et al., 2024; Lipman et al., 2023; Oquab et al., 2024). Generative self-alignment removes the external encoder and instead constructs targets from the generator itself. Self-Representation Alignment (SRA) (Jiang et al., 2025) and Self-Flow (Chefer et al., 2026) obtain these targets from a momentum teacher, an exponential moving average (EMA) of the trainable network. We ask whether this temporal averaging is necessary for effective self-alignment in pixel-space generation.

Prior findings in representation learning and latent-space generation do not establish whether pixel-space self-alignment requires an EMA teacher. Momentum teachers are widely used in self-supervised learning (He et al., 2020; Grill et al., 2020; Caron et al., 2021), yet SimSiam (Chen & He, 2021) learns useful representations with shared weights, a predictor, and a stop-gradient target. Generative self-alignment additionally receives a flow-matching signal, making the joint training objective different. SRA's latent-space ablation favors EMA over online targets (Jiang et al., 2025,

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.

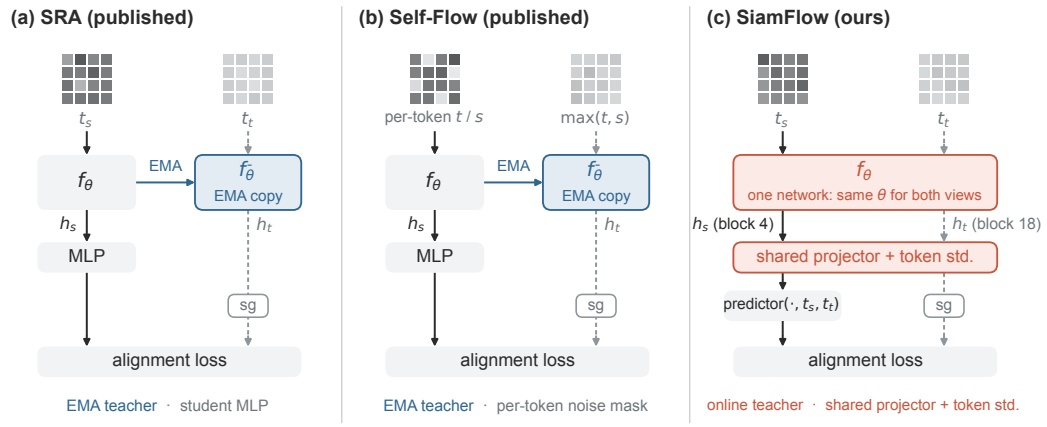


Figure 2: **Self-alignment with EMA and online teachers.** SiamFlow uses an online teacher, shared projector, token standardization, and student predictor. Dashed paths stop gradients. JiT adaptations: Table 1.

Tab. 6); whether the same choice is needed for a generator trained directly on pixels remains an empirical question.

We study this question in JiT (Li & He, 2025) and introduce *SiamFlow* (Figures 1 and 2). The current generator processes both noisy views, with shallow student features aligned to deeper features from a cleaner target view. A shared projector maps both branches into the alignment space, and a student-only predictor learns to match the stop-gradient target. To compare online and EMA teachers, we change only the parameter source of the teacher backbone, keeping the projector shared and the remaining training settings fixed. All models retain EMA weights for sampling; the comparison concerns the teacher used during training.

On ImageNet 256×256 with JiT-L/16, SiamFlow achieves FID 2.13, compared with 2.33 for the same recipe with an EMA teacher and 2.39 without alignment. Online targets also improve our JiT adaptation of SRA, from 2.36 to 2.24. SiamFlow improves the JiT baseline at larger model size and image resolution, and approaches PixelREPA's reported FID of 2.11 with the same 459M-parameter backbone (Shin et al., 2026), without requiring its external semantic encoder (Table 3). These results establish the potential of the current generator's features to provide effective supervision for pixel-space generation.

We next investigate how to make this online supervision effective. The alignment head matters: an MLP without variance control exhibits large FID oscillations, while adding a variance penalty improves online FID enough to outperform the corresponding EMA model. For the convolutional projector, token standardization has the larger effect. Decomposing standardization identifies feature rescaling as the main source of improvement, and depth ablations favor supervising an early student layer with a deeper intermediate target. Across the tested JiT configurations, the relative FID performance of online and EMA teachers depends on the alignment recipe.

Noise augmentation also warrants reassessment. Jiang et al. (2026) attribute the gains of dual-timestep inputs primarily to data augmentation in latent-space generation. In our JiT adaptation of Self-Flow, removing token-level noise masking instead improves FID by 0.18 on L/16 and 0.23 on B/16 (Table 7). This challenges the transferability of that augmentation benefit to pixel-space self-alignment and supports revisiting inherited training choices alongside the teacher design.

Finally, a fixed-target analysis distinguishes the conditional image mean predicted by flow matching from the conditional feature direction predicted by alignment. Faster EMA updates reduce target and alignment-gradient differences but do not close the FID gap. We also ask whether stronger cross-noise instance retrieval accompanies better generation: EMA-trained models achieve higher retrieval accuracy in most comparisons, while online-trained models have lower FID at every evaluated checkpoint (Section 6.3).

Contributions.

- We introduce SiamFlow and challenge the reliance on EMA teachers in generative self-alignment. Online targets achieve FID 2.13 in pixel-space JiT without an external semantic encoder and outperform EMA targets in two matched alignment recipes (Section 4).
- We investigate the design of online self-alignment through controlled ablations of variance control, feature rescaling, and student–teacher depth. We also show that removing token-level noise masking improves our Self-Flow adaptation at two model sizes, challenging the transferability of its reported latent-space benefits (Section 5 and Appendix B).
- We examine the role of EMA beyond generation scores: faster EMA updates reduce target differences without matching online FID, and higher cross-noise retrieval accuracy does not consistently predict better generation quality (Section 6).

2 Related work

Representation alignment. REPA (Yu et al., 2025a) aligns a diffusion transformer's intermediate features to a pretrained encoder and improves FID on DiT and SiT (Peebles & Xie, 2023; Ma et al., 2024). iREPA (Singh et al., 2026) attributes the gain to the spatial structure of the target features, which motivates examining spatial projectors in generative self-alignment. RiT (Zhang et al., 2026) generates directly in the feature space of a frozen DINOv2 encoder and maps the features to pixels with a pretrained decoder. PixelREPA (Shin et al., 2026) reports that REPA applied directly to JiT (Li & He, 2025) degrades FID and repairs it with a shallow transformer adapter and a masked input. Self-alignment removes the external encoder: SRA (Jiang et al., 2025) aligns a noisier view to a cleaner view of the same model, Self-Flow (Chefer et al., 2026) adds a per-token dual-timestep mask, and Jiang et al. (2026) extend the family. All three take the target from an EMA copy.

Self-supervised learning and collapse prevention. BYOL (Grill et al., 2020) uses a momentum encoder, while SimSiam (Chen & He, 2021) showed that a stop-gradient and a predictor suffice without it. Tian et al. (2021) analyze the predictor's role, and Halvagal et al. (2023) show that predictor and target together act as an implicit variance regularizer. Explicit alternatives include variance regularization in VICReg (Bardes et al., 2022), the cross-correlation penalty of Barlow Twins (Zbontar et al., 2021), batch whitening in W-MSE (Ermolov et al., 2021), and centering and sharpening in DINO (Caron et al., 2021), which Wu et al. (2025) replace with a coding-rate term. Token standardization is a form of affine-free instance normalization (Ulyanov et al., 2016). Richemond et al. (2020) keep BYOL's momentum encoder and change the normalization; our teacher comparisons keep the projector fixed and change the teacher.

Pixel-space generation. JiT, PixelFlow, PixNerd, EPG, and PixelDiT generate directly in pixel space (Li & He, 2025; Chen et al., 2025; Wang et al., 2025; Lei et al., 2025; Yu et al., 2025b). By learning directly from pixels, these models support end-to-end generative training without relying on a separately pretrained image tokenizer or its fixed latent space. This avoids inheriting the tokenizer's reconstruction errors and representational constraints, and offers a promising path for scaling generation directly with model capacity and data. These advantages motivate our study of self-alignment in pixel space. We use JiT to isolate teacher choice within matched training recipes; Table 3 provides broader comparisons.

3 SiamFlow: self-alignment with an online teacher

SiamFlow aligns features from differently noised views using an online teacher (Figure 2c). The teacher branch, projector, token standardization, and predictor are used only during training; sampling uses the generator alone, with no extra modules. Figure 3 shows the computation graph.

Noisy views and features. For an image x_0, we sample $t_s < t_t$ and independent Gaussian noises ϵ_s, ϵ_t, and form $z_s = t_s x_0 + (1 - t_s)\epsilon_s$ and $z_t = t_t x_0 + (1 - t_t)\epsilon_t$; larger t denotes a cleaner input. The student feature h_s is taken after block 4 of the 24-block JiT-L/16 backbone and the target feature h_t after block 18 (block indices are one-based). The student forward continues through the remaining blocks and gives the flow-matching prediction.

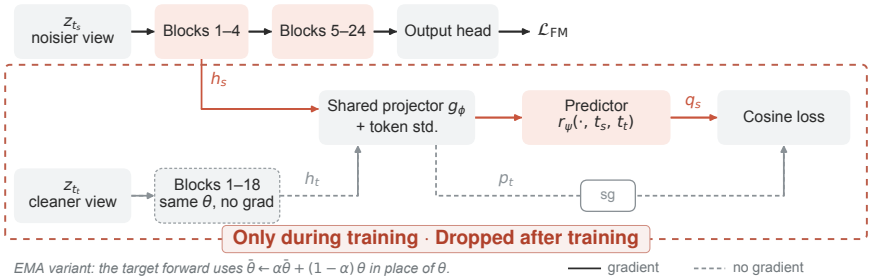


Figure 3: **SiamFlow training graph.** Both views share the backbone and projector; targets are stop-gradient. The alignment head is discarded after training.

Projector and token standardization. A shared projector g_ϕ maps the student and teacher backbone features h_s and h_t to the alignment space. In SiamFlow, it consists of a 1×1 convolution, a GELU activation, a 3×3 convolution, and Layer Normalization (LN), with hidden and output width $D = 1024$. LN normalizes the channels within each token.

Token standardization additionally normalizes each channel across the spatial tokens of an image. Let $h \in \mathbb{R}^{B\times N\times C}$ denote either h_s or h_t, where B is the per-device batch size, $N = 256$ is the number of spatial tokens, and C is the backbone feature dimension. The projected features are $u = g_\phi(h) \in \mathbb{R}^{B\times N\times D}$; u_{bnd} denotes channel d of token n in image b. We compute the spatial mean μ_{bd} and standard deviation s_{bd} for each image and channel, and define the standardization operator S as

$$\mu_{bd} = \frac{1}{N}\sum_n u_{bnd}, \qquad s_{bd}^2 = \frac{1}{N-1}\sum_n (u_{bnd} - \mu_{bd})^2, \qquad S(u)_{bnd} = \frac{u_{bnd} - \mu_{bd}}{\max(s_{bd}, 10^{-4})}. \tag{1}$$

Applying S to both branches gives the student projections $p_s = S(g_\phi(h_s))$ and teacher projections $p_t = S(g_\phi(h_t))$. A student-only predictor $q_s = r_\psi(p_s, t_s, t_t)$ maps the student projections to the teacher's alignment space. It consists of three linear layers separated by two GELU activations and is conditioned on both timesteps. The asymmetric predictor and stop-gradient teacher branch are inspired by SimSiam (Chen & He, 2021).

Online teacher. The teacher uses the online backbone f_θ and the shared projector g_ϕ:

$$h_t = f_\theta^{(18)}(z_t, t_t), \qquad p_t = \text{sg}[S(g_\phi(h_t))]\,.$$

Here $f_\theta^{(18)}$ denotes the block-18 feature map (class conditioning omitted), and sg stops gradients. Only the student branch uses the predictor r_ψ.

Training objective. Flow matching supervises image prediction, while cosine alignment supervises intermediate features. Following VICReg (Bardes et al., 2022), we include a variance regularizer $\mathcal{V}$ that penalizes feature channels with standard deviation below one. We evaluate its effect separately from token standardization in Section 5.1. The full objective is

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{FM}} - \frac{\lambda}{BN}\sum_{b,n}\cos(q_{s,bn}, \text{sg}[p_{t,bn}]) + \lambda_{\text{var}}\mathcal{V}(p_s), \\ \mathcal{V}(p) &= \frac{1}{D}\sum_d \max\left(0, 1 - \sqrt{\text{Var}_{b,n}(p_d) + 10^{-4}}\right), \end{aligned} \tag{2}$$

Here $\lambda = 0.5$, $\lambda_{\text{var}} = 1$, and $\text{Var}_{b,n}$ is the sample variance over the batch's BN tokens. Token standardization already fixes spatial variance, so $\mathcal{V}$ has zero upstream gradient in exact arithmetic when the numerical clamp is inactive (Appendix F). Standardization thus supplies the variance constraint in SiamFlow; Section 5.1 isolates the penalty's effect.

3.1 A CONDITIONAL-EXPECTATION VIEW

The flow-matching term $\mathcal{L}_{\text{FM}}$ and the cosine-alignment term in Equation (2) train predictions from the same noisy input z_s. For JiT's clean-image prediction F, the flow-matching loss is a timestep-weighted squared error to x_0 (Appendix E). At fixed timestep and class condition c, its minimizer is

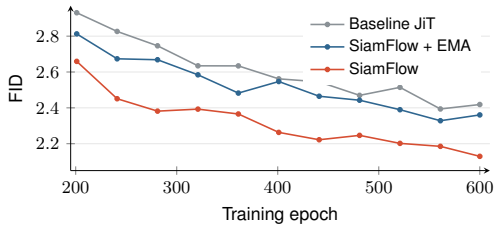


Figure 4: **Online targets improve FID throughout training.** JiT-L/16, 256^2, epoch 201 onward; all models use sampling EMA.

Table 2: **Online vs. EMA teachers.** JiT-L/16, 256^2, 600 epochs. Within each recipe, only the teacher-backbone weights change; the projector remains shared.

Recipe	Teacher	FID↓	IS↑
Baseline JiT	—	2.39	293.8
SiamFlow recipe	EMA	2.33	292.5
SiamFlow	online	**2.13**	291.6
SRA recipe	EMA	2.36	292.3
SRA recipe	online	2.24	294.2

$F^\star = \mathbb{E}[x_0 \mid z_s, t_s, c]$. For the alignment term, with the target map fixed, an idealized predictor, and independent noise in the two views, the expected negative cosine is minimized by

$$\hat{q}_s^\star = \frac{m}{\|m\|}, \qquad m = \mathbb{E}\big[\bar{U}(x_0) \mid z_s, t_s, t_t, c\big], \quad \bar{U}(x_0) = \mathbb{E}[U \mid x_0, t_t, c], \tag{3}$$

for $m \neq 0$, where U is the normalized target feature of one token (Appendix E). The two losses thus have different ideal prediction targets: a conditional image mean and a direction of conditional mean features. This characterization holds for a fixed target map; it does not determine which teacher yields better generation. We examine retrieval and FID empirically in Section 6.3. Separately, for a fixed prediction at angle $\varphi \in (0, \pi)$ from the target, moving the target by an angle δ changes the gradient at the prediction by at most $2\sin(\delta/2)/\sin\varphi$ in relative terms, approximately $\delta/\sin\varphi$ for small δ in radians (Section 6.2).

4 Online Self-Alignment for Image Generation

We evaluate whether an online teacher can improve pixel-space generation, first within matched alignment recipes and then in comparison with prior methods.

4.1 Experimental Setup

We train JiT-L/16 on ImageNet 256×256 (Deng et al., 2009) for 600 epochs with global batch size 1024 and AdamW (Loshchilov & Hutter, 2017). Sampling uses a 50-step Heun solver (Karras et al., 2022) with classifier-free guidance 2.4 (Ho & Salimans, 2021), and FID is computed over 50,000 images (Heusel et al., 2017).

Table 1: **JiT-L/16 training recipes.** SRA/Self-Flow are our adaptations. TS: token standardization; Pred.: predictor; Noise: shared or independent; Gap: timestep-gap cap; Mask: token-mask ratio.

Recipe	Teacher	Projector	TS	Pred.	Blocks	λ	λ_{var}	Noise	Gap	Mask
SRA	EMA	shared MLP + LN	no	yes	$6 \to 16$	0.2	0	shared	0.2	no
Self-Flow	EMA	student MLP	no	no	$7 \to 17$	0.8	0	shared	—	0.25
SiamFlow	online	shared Conv + LN	yes	yes	$4 \to 18$	0.5	1	indep.	1.0	no

Comparison recipes. To compare teacher choices within a common pixel-space framework, we implement SiamFlow, SRA, and Self-Flow in JiT under the evaluation protocol above. Table 1 summarizes their alignment settings. We refer to the adapted methods as the SRA and Self-Flow recipes. The SRA recipe follows SRA's layer ratio, alignment weight, shared noise draw, and timestep-gap cap, but uses the shared MLP + LN projector and a student predictor, whereas published SRA trains a student-only MLP. The Self-Flow recipe follows the official implementation, including its per-token noise mask. SiamFlow uses the same noise level for all tokens of a view: in the Self-Flow recipe, removing the mask lowers FID by 0.18 at L/16 and by 0.23 at B/16 (Appendix B).

4.2 Online vs. EMA Teachers

We compare online and EMA weights for the teacher backbone, keeping the projector shared between student and teacher and all other training settings fixed.

Table 3: **ImageNet generation with classifier-free guidance.** Left: published results. Right: our 600-epoch runs and PixelREPA. †Published; protocols differ. §JiT adaptation with EMA. Parameters exclude external encoders/autoencoders. Bold: best FID without an external encoder per block.

Method	Ep.	Params	FID↓	IS↑	Method	Params	FID↓	IS↑
Latent space, 256 × 256					*JiT-L/16,* 256 × 256			
DiT-XL/2†	1400	675M	2.27	278.2	Baseline JiT	459M	2.39	293.8
SiT-XL/2†	1400	675M	2.06	270.3	SRA recipe§	459M	2.36	292.3
REPA, SiT-XL/2†	800	675M	1.42	305.7	Self-Flow recipe§	459M	2.46	266.8
Pixel space, 256 × 256					SiamFlow + EMA teacher	459M	2.33	292.5
PixelFlow-XL†	320	677M	1.98	282.1	SiamFlow	459M	**2.13**	291.6
PixNerd-XL†	320	700M	1.93	298.0	PixelREPA†	459M	2.11	309.5
EPG-XXL/16†	600	789M	1.81	294.6	*JiT-XL/16,* 256 × 256			
PixelDiT-XL†	320	797M	1.61	292.7	Baseline JiT	676M	2.07	284.4
JiT-G/16†	600	2B	1.82	292.6	SiamFlow	676M	**1.96**	307.0
JiT-L/16†	—	459M	2.36	298.5	*JiT-L/32,* 512 × 512			
Representation space, 256 × 256					Baseline JiT	462M	2.67	299.1
RiT-XL†	800	676M	1.14	299.7	SiamFlow	462M	**2.46**	309.1

SiamFlow achieves FID 2.13, compared with 2.33 for the same recipe with an EMA teacher and 2.39 for Baseline JiT (Table 2). Figure 1 compares alignment recipes, while Figure 4 tracks the matched SiamFlow runs over the last 400 epochs; SiamFlow is lower at each of these 11 evaluations, by 0.12 to 0.29. This ordering persists across the measured training stages.

The same comparison in our JiT adaptation of SRA tests whether the benefit extends beyond SiamFlow. This configuration uses an MLP + LN projector without variance regularization or token standardization (Table 1). With its EMA teacher this recipe obtains 2.36, close to the baseline's 2.39; the online teacher lowers FID to 2.24 (Table 2). The online advantage therefore persists without SiamFlow's projector and standardization choices. Both teacher variants in our SRA adaptation use the same shared projector and student-only predictor.

4.3 Comparison with prior methods

Table 3 places SiamFlow alongside self-alignment methods and generators operating in pixel, latent, and representation spaces. SiamFlow improves the JiT baseline across the tested model sizes and resolutions: FID decreases from 2.39 to 2.13 with JiT-L/16, from 2.07 to 1.96 with JiT-XL/16, and from 2.67 to 2.46 with JiT-L/32 at 512 × 512. These gains show that self-alignment can improve pixel-space generation using supervision from the generator itself. The XL/16 and 512 × 512 runs compare alignment with no alignment; teacher comparisons are restricted to L/16 at 256 × 256. Additional evaluation details are in Appendix G.

A substantial gap remains to stronger latent- and representation-space systems in the table. REPA with SiT-XL/2 achieves FID 1.42 (Yu et al., 2025a), and RiT-XL achieves 1.14 (Zhang et al., 2026), compared with SiamFlow-XL/16's 1.96. Our main focus is the comparison with existing self-alignment approaches under a common JiT backbone. At JiT-L/16, SiamFlow achieves 2.13, compared with 2.36 for our SRA adaptation and 2.46 for our Self-Flow adaptation, both using EMA teachers. It also approaches PixelREPA's reported 2.11 with the same 459M-parameter JiT-L/16 backbone (Shin et al., 2026). PixelREPA uses DINOv2 as an external semantic encoder, whereas SiamFlow obtains its alignment supervision entirely from the online generator. This comparison highlights the potential of self-alignment to approach the quality of external-encoder supervision without requiring a pretrained semantic teacher.

5 Investigating design choices for online self-alignment

The preceding results show that online self-alignment improves generation quality. In this section, we investigate how projector design, feature normalization, and alignment depth contribute to these gains and affect training stability.

5.1 Matching variance control to the projector

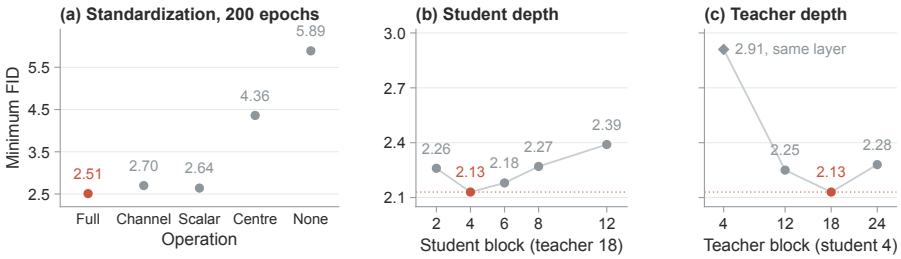


Figure 5: **Standardization and alignment-depth ablations.** (a) MLP + LN, 200 epochs; (b,c) Conv + LN + TS, 600 epochs. Color: SiamFlow; diamond: same-layer alignment. Each depth sweep fixes the other branch. Definitions and values: Tables 8 and 9.

The two variance controls help different projectors; combining them does not consistently improve FID. The MLP benefits from a variance penalty, whereas the SiamFlow convolutional projector benefits from token standardization (Table 4). This interaction is the main design lesson: the projector and its variance control should be chosen together.

For the MLP, the variance penalty provides the gain. Adding it lowers FID from 2.45 to 2.18; adding token standardization on top gives 2.23, with no further improvement. The unregularized run also exhibits large FID oscillations, examined in Section 6.1. This MLP applies two linear layers with an intervening GELU and final LN, using hidden and output width 1024.

Table 4: **Online-teacher ablations.** JiT-L/16, 600 epochs; TS: token standardization.

Projector	TS	λ_{var}	FID↓
MLP	no	0	2.45
MLP	no	1	2.18
MLP	yes	1	2.23
Conv 3×3	no	1	2.25
Conv 3×3	yes	0	2.15
Conv (SiamFlow)	yes	1	**2.13**
SiamFlow with one training change			
Shared noise draw for both views			2.15
Timestep-gap cap 0.2			2.19
Predictor removed			2.24

For the convolutional projector, token standardization has the larger effect on FID. Removing it raises FID from 2.13 to 2.25, whereas removing the variance penalty gives 2.15. The observed difference is therefore larger for standardization. Under token standardization, the variance penalty has zero upstream gradient in exact arithmetic (Appendix F), so the 0.02 difference alone does not establish an additional benefit from the penalty.

5.2 Identifying the role of feature rescaling

Token standardization combines two operations: subtracting the mean and rescaling by the standard deviation. We separate them to ask whether the improvement comes from centering, from controlling feature scale, or specifically from normalizing each channel separately. All five variants use the same MLP + LN projector, online teacher, and 200-epoch budget; they differ only in the operation applied to the projector's LN output.

Let u denote these features, μ_d and s_d their mean and standard deviation across spatial tokens for channel d of each image, and s_{all} a single standard deviation across all tokens and channels of that image. The labels in Figure 5a denote **Full**: centering and per-channel rescaling, $(u - \mu_d)/s_d$; **Channel**: per-channel rescaling without centering, u/s_d; **Scalar**: one shared rescaling factor per image, u/s_{all}; **Centre**: centering without rescaling, $u - \mu_d$; and **None**: no transformation after LN, u. All rescaling denominators are clamped at 10^{-4}.

Rescaling produces the main improvement; centering alone is insufficient. None gives FID 5.89, and Centre improves it to 4.36. Both Channel (2.70) and Scalar (2.64) perform substantially better, while Full gives the lowest FID (2.51). Thus, much of the benefit is already obtained by adjusting feature scale, even with one scalar per image. Full standardization achieves the lowest FID among the five variants.

A possible explanation is reduced sensitivity to feature scale at the predictor input. For a positive scale factor c, centering retains the scale change: $cu - \mu(cu) = c(u - \mu(u))$. Rescaling cancels it: $cu/s(cu) = u/s(u)$ when the denominator clamp is inactive. Unlike the cosine loss at its output, the GELU-based predictor is not scale-invariant, so changes in its input magnitude can alter its

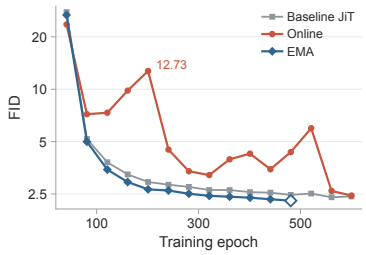


Figure 6: **FID without variance control.** MLP + LN, no TS or variance penalty. EMA stopped at epoch 497 (hollow marker).

Table 5: **Teacher choice and head design.** JiT-L/16; shared projector. TS: token standardization. *Stopped at epoch 497.

			FID↓	
Projector	TS	λ_{var}	Online	EMA
MLP	no	0	2.45	**2.29***
MLP	no	1	**2.18**	2.27
MLP	yes	1	**2.23**	2.37
Conv 3×3	yes	1	**2.13**	2.33

activations and gradients. We hypothesize that suppressing such input-scale variation makes the alignment mapping easier to learn.

5.3 Choosing student and teacher depths

We vary student and teacher depths separately, fixing the other at block 18 and block 4, respectively (Figure 5b,c). All runs use JiT-L/16 with the SiamFlow head for 600 epochs. **The best tested setting supervises an early student layer with a deeper intermediate target.** Blocks $4 \rightarrow 18$ give FID 2.13, compared with 2.91 for same-layer alignment ($4 \rightarrow 4$) and 2.39 when supervision moves to student block 12. Using the final teacher block gives 2.28, so a deeper target is not always better. These sweeps motivate SiamFlow's $4 \rightarrow 18$ choice (Table 9).

6 Examining the role of the EMA teacher

The preceding experiments establish the effectiveness of online alignment. Here we examine how variance regularization and EMA decay affect the teacher comparison, and whether better cross-noise retrieval accompanies better generation.

6.1 Improving online alignment with variance regularization

Without variance control, online alignment can become unstable during training. With the MLP + LN projector and neither variance regularization nor token standardization, FID fluctuates sharply instead of improving steadily (Figure 6). The EMA teacher produces a smoother trajectory under the same settings. This comparison changes only the teacher backbone from online to EMA weights; the projector remains shared between student and teacher.

Variance regularization improves online alignment enough to outperform the corresponding EMA model. Adding the penalty lowers online FID from 2.45 to 2.18, compared with 2.27 for EMA under the same regularized configuration (Table 5). With token standardization added, online also achieves lower FID (2.23 versus 2.37). In this MLP configuration, EMA achieves lower FID without the variance penalty, whereas online achieves lower FID with it. Our SRA adaptation also favors online targets without this penalty (Section 4.2), so variance regularization alone does not determine which teacher achieves lower FID.

6.2 Does a faster EMA teacher close the FID gap?

An EMA teacher averages past backbone weights and therefore follows the current network with a delay. Could this delay explain its higher FID? We test a faster EMA update by reducing the decay from 0.9999 to 0.99, shortening the approximate averaging window from 10,000 to 100 steps while keeping the remaining SiamFlow settings fixed.

Table 6: **Faster EMA leaves a FID gap.** Target angles and relative block 1–4 gradient changes at checkpoints 100–500; FID over full runs.

Teacher	Angle	Grad. change	FID↓
EMA 0.9999	3.1–3.6°	0.16–0.35	2.33
EMA 0.99	1.1–1.9°	0.07–0.19	2.29
Online	—	—	**2.13**

We first check whether the faster EMA produces targets closer to the online network. At saved checkpoints of each EMA run, we compute both target sources on the same inputs while holding the student and shared projector fixed. We measure the angle between the projected, standardized targets and the relative change in the alignment gradient for student blocks 1–4. The gradient change is $\|g_{\text{online}} - g_{\text{EMA}}\|/\|g_{\text{EMA}}\|$; Appendix E gives the measurement details.

Reducing EMA decay to 0.99 narrows the target and gradient differences, but its FID remains above the online teacher's. FID changes from 2.33 to 2.29, compared with 2.13 for online targets (Table 6). The shorter averaging window brings the teacher closer to the current network without closing the generation-quality gap.

6.3 Better SSL-style retrieval does not imply better generation

Inspired by instance discrimination (Wu et al., 2018), we use cross-noise retrieval to measure image-identity preservation and ask whether higher accuracy accompanies better generation. Retrieval matches independently noised views of $1{,}024$ images using pooled backbone features; Recall@1 measures how often the top match is the same image. We compare online- and EMA-trained models at five checkpoints from 100 to 500, both evaluated with sampling-EMA weights.

EMA-trained models retrieve better, while online-trained models generate better. Across blocks $4, 12, 18, 24$ and the five checkpoints, EMA has higher Recall@1 in $19/20$ comparisons, whereas online has lower FID at all five checkpoints (Figure 7). Mean subtraction preserves the $19/20$ result; with class conditioning at checkpoint 500, EMA also has higher Recall@1 at all four depths. Thus, cross-noise retrieval accuracy is not a reliable proxy for generation FID in these comparisons. Protocols and full control results are in Appendix D.

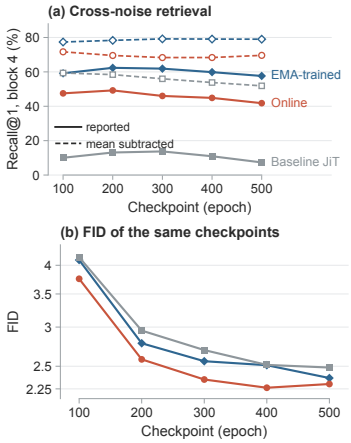


Figure 7: **Higher retrieval, worse FID.** Top: block-4 Recall@1 (solid: original; dashed: mean-subtracted). Bottom: FID. All use sampling EMA.

7 Conclusion

We introduced SiamFlow, which aligns shallow student features with deeper targets from a cleaner view using the current generator. On ImageNet with JiT-L/16, it achieves FID 2.13, improving on the baseline and EMA-teacher counterpart and approaching PixelREPA's 2.11 without an external encoder. Gains extend to larger models and higher resolution. The teacher comparison also depends on the alignment recipe: online targets achieve lower FID in SiamFlow and our SRA adaptation, while EMA performs better in the unregularized MLP configuration. Faster EMA reduces target differences without closing the FID gap. Cross-noise retrieval and generation favor different models, so this retrieval metric cannot substitute for direct generation evaluation. Removing token-level masking also improves our Self-Flow adaptation at two model sizes. These findings motivate revisiting inherited alignment choices and show that momentum teachers are not necessary in the tested pixel-space settings. SiamFlow supplies supervision from the generator itself without adding modules at sampling.

Reproducibility Statement. Appendix A lists the backbone configurations, the optimizer and schedule, the projector and predictor architectures, the timestep-pair sampling, the loss weights, the sampler and guidance settings, the FID protocol, and the hyperparameters of the SRA and Self-Flow recipes. It also records the runs that stopped before completing their schedule; their values are best-so-far FIDs and are marked wherever they appear. We will release the code, the configuration files, and the trained model weights.

AI Use Statement. In this work, we used large language models to help write code, to review drafts of the paper, to edit paragraphs, and to improve grammar. We have reviewed all AI-assisted work, including the code and every edited passage. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *Preprint*, 2026. https://bfl.ai/research/self-flow.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, 2021.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Manu Srinath Halvagal, Axel Laborieux, and Friedemann Zenke. Implicit variance regularization in non-contrastive SSL. In *Advances in Neural Information Processing Systems*, 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025.

Dengyang Jiang, Mengmeng Wang, Harry Yang, and Jingdong Wang. From SRA to Self-Flow: Data augmentation or self-supervision? *arXiv preprint arXiv:2607.02508*, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jiachen Lei, Keli Liu, Julius Berner, Haiming Yu, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. Advancing end-to-end pixel space generative modeling via self-supervised pre-training. *arXiv preprint arXiv:2510.12586*, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision (ECCV)*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy 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 Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Pierre H. Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, and Michal Valko. BYOL works even without batch statistics. In *NeurIPS Workshop on Self-Supervised Learning*, 2020. arXiv:2010.10241.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

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 *The Fourteenth International Conference on Learning Representations*, 2026.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, 2021.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Ziyang Wu et al. Simplifying DINO via coding rate regularization. *Preprint*, 2025.

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 *International Conference on Learning Representations (ICLR)*, 2025a.

Yongsheng Yu, Wei Xiong, Weili Nie, et al. PixelDiT: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025b.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 2021.

Le Zhang, Ning Mang, and Aishwarya Agrawal. RiT: Vanilla diffusion transformers suffice in representation space. *arXiv preprint arXiv:2605.21981*, 2026.

A IMPLEMENTATION DETAILS

This section specifies the models, training objectives, and evaluation protocol used in the main experiments. Code and configuration files will be released.

Backbone. JiT (Li & He, 2025), a diffusion transformer on RGB with no autoencoder. Depth / width / heads / patch: B/16 12 / 768 / 12 / 16; L/16 24 / 1024 / 16 / 16; L/32 24 / 1024 / 16 / 32; XL/16 28 / 1152 / 16 / 16; the latent control is L/2, 24 / 1024 / 16 / 2. The FFN is SwiGLU at `mlp_ratio` 4, normalization is RMSNorm with QK-norm, and position information is a fixed 2-D sin-cos embedding plus RoPE. Conditioning enters as 32 in-context tokens, repeated class embeddings with learned position embeddings, inserted from block 8 at L/16 (block 4 at B/16, block 9 at XL/16). There are no dedicated register tokens. The patch embedding is a two-stage bottleneck convolution with bottleneck width 128.

Optimization. AdamW (Loshchilov & Hutter, 2017), based on Adam (Kingma & Ba, 2014), with $\beta = (0.9, 0.95)$, PyTorch default ϵ. Base learning rate 5×10^{-5}, scaled as blr $\times$ global batch$/256$, giving 2.0×10^{-4} at global batch 1024. Constant schedule after a linear warmup of 5 epochs applied per iteration. Weight decay is 0. Gradient clipping 1.0 at B/16, L/16, XL/16 and in latents; disabled in the 512×512 runs. `bfloat16` autocast with the attention $QK^\top$ product forced to fp32. Eight GPUs, per-GPU batch 128, global batch 1024, 600 epochs. Label dropout 0.1. Projection dropout 0 at L/16 and L/32, 0.2 at XL/16. Gradient checkpointing is on for the XL/16 and Self-Flow runs and off for the L/16 runs reported here.

EMA teacher implementation. The teacher backbone is updated after each optimization step as $\bar{\theta} \leftarrow \alpha\bar{\theta} + (1-\alpha)\theta$, with $\alpha = 0.9999$ unless otherwise specified. Backbone buffers are copied from the student. The projector remains shared, the predictor remains student-only, and teacher features are computed without gradients. Neither the projector nor the predictor has a separate EMA copy.

Sampling weights. Every model is evaluated using sampling-EMA weights with decay 0.9999, including models trained with an online teacher. Sampling EMA is distinct from the teacher backbone used for alignment. The online teacher uses the current student backbone without gradients; the EMA teacher uses the averaged backbone described above.

Flow parameterization and the timestep pair. $z_\tau = \tau x_0 + (1-\tau)\epsilon$ with velocity target $(x_0 - z_\tau)/\max(1-\tau, \tau_{\min})$ and $\tau_{\min} = 5\times10^{-2}$. Timesteps are drawn as $\tau = \sigma(u)$, $u \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$ with $P_{\text{mean}} = -0.8$ and $P_{\text{std}} = 0.8$. For a pair, t_s is drawn from that distribution, then gap $\sim U(0,1) \cdot \min(t_{\text{gap}}^{\max}, 1 - t_s)$ and $t_t = \min(t_s + \text{gap}, 1)$. The default $t_{\text{gap}}^{\max}$ is 1.0, the full range; the ablation row in Table 4 sets it to 0.2. The two views take independent noise draws by default; the label-dropout mask is drawn once and shared between them.

Projection head. The shared convolutional projector consists of Conv2d$(1024 \to 1024, k = 1)$, GELU, Conv2d$(1024 \to 1024, k = 3, \text{pad} = 1)$, and LayerNorm$(1024)$. The alternative MLP is Linear, GELU, Linear, LayerNorm, with width 1024. LN normalizes channels within each token; token standardization normalizes spatial tokens within each image and channel. Across-token variance can decrease even when LN is present. The predictor is downstream of both operations, so its output norm is a separate statistic. The standardization is computed in fp32 with the standard deviation clamped at 10^{-4}. The predictor runs on the student side only: `Linear`$(1024{+}512 \to 1024)$, GELU, `Linear`$(1024 \to 1024)$, GELU, `Linear`$(1024 \to 1024)$, where the 512 is the concatenation of two 256-dimensional sinusoidal timestep embeddings of t_s and t_t, broadcast over tokens. The alignment loss is a per-token negative cosine with normalization $\epsilon = 10^{-6}$, weight $\lambda = 0.5$. The variance term takes the standard deviation over the flattened $(B{\cdot}N, D)$ projections as $\sqrt{\text{var} + 10^{-4}}$ and applies $\text{ReLU}(\gamma - \text{std})$ with $\gamma = 1$ and $\lambda_{\text{var}} = 1$.

Sampling and evaluation. Heun (Karras et al., 2022) with 50 steps on a uniform grid in τ, the last segment Euler. Classifier-free guidance (Ho & Salimans, 2021) at 2.4 for L/16 at 256×256, 2.9 for B/16, 2.2 for XL/16, 2.5 for L/32 at 512×512, and 2.4 for the L/2 latent pair, applied on the interval $\tau \in (0.1, 1.0)$ and set to 1.0 outside it. FID (Heusel et al., 2017) and Inception Score (Salimans et al.,

2016) over 50,000 samples with equal counts per class, against precomputed reference statistics for the matching resolution.

Data. ImageNet-1k (Deng et al., 2009) train split, ADM center crop, random horizontal flip, pixels scaled to $[-1, 1]$. No other augmentation. The latent control encodes with `stabilityai/sd-vae-ft-ema`, scale factor 0.18215, taking a posterior sample rather than the mode.

Adaptations of prior work. *SRA recipe* (Jiang et al., 2025, App. C and App. F): layer pair $6 \to 16$, $\lambda = 0.2$, no variance term, EMA teacher at 0.9999, shared noise draw between the two views, $t_{\text{gap}}^{\max} = 0.2$, shared MLP + LN projector, predictor retained. Published SRA uses a student-only trainable MLP with no teacher projector (Jiang et al., 2025, Fig. 3). Our adaptation uses the shared MLP + LN projector and a separate student predictor defined above. The layer pair, the loss weight and the noise draw are read from the official implementation rather than from the published description. The online control in Section 4.2 uses current rather than EMA backbone weights for the target, retaining the shared projector and all other settings. *Self-Flow recipe* (Chefer et al., 2026, App. A.3): the official head, with student and teacher blocks set by depth ratios 0.3 and 0.7, so blocks 7 and 17 at depth 24; mask ratio 0.25; representation loss weight 0.8; teacher EMA 0.9999; one noise draw shared between the two views. Per-token timesteps are drawn by sampling t and s independently from $p(\tau)$ and setting $\tau_n = s$ on masked tokens and t elsewhere, with no ordering imposed, while the teacher sees a uniform input at the cleaner of the two levels (Chefer et al., 2026, §3.4), which in our convention is $\max(t, s)$. The remaining details — that the per-token level enters through adaLN, and that the official projector is `Linear`, SiLU, `Linear`, with no projector on the teacher side, no predictor and no variance term — are read from the official implementation rather than from the source paper. The mask comparison evaluates this JiT adaptation under matched settings.

B Testing token-level noise masking in pixel space

Self-Flow applies two noise levels to different tokens of the same student input. We test whether this augmentation helps our JiT adaptation by comparing masked and uniform inputs at two model sizes, keeping the EMA teacher, head, and evaluation settings fixed within each comparison.

Table 7: **Input-noise masking with a fixed EMA teacher.** Minimum FID at two model sizes, with each adaptation compared to Baseline JiT of the same size. Guidance is 2.4 at L/16 and 2.9 at B/16, with matched mask comparisons within each model size.

	JiT-L/16 FID↓	JiT-B/16 FID↓
Baseline JiT	2.39	3.57
Self-Flow recipe	2.46	3.74
Self-Flow recipe, mask removed	2.28	3.51
mask off − mask on	−0.18	−0.23
Self-Flow recipe − Baseline JiT	+0.06	+0.17

Removing the mask lowers FID from 2.46 to 2.28 on L/16 and from 3.74 to 3.51 on B/16 (Table 7). Thus, token-level masking is not beneficial in these pixel-space comparisons. The experiment tests the input-noise mask; it does not test attention separation.

C Detailed standardization and depth ablations

This section provides the values underlying Figure 5. The standardization experiment isolates centering and rescaling using the same MLP + LN projector and 200-epoch budget. Rescaling gives most of the FID improvement, and full standardization performs best (Table 8). The depth experiment holds one branch's feature depth fixed while varying the other for 600 epochs. Blocks $4 \to 18$ give the lowest FID in this sweep (Table 9).

Table 8: **Standardization components at 200 epochs.** MLP + LN with online targets; six evaluations per configuration. Here u is the LN output, μ_d, s_d are its per-image channel statistics across tokens, and s_{all} is one standard deviation across all features of an image. Denominators are clamped at 10^{-4}.

Operation	Output	FID↓	Std., start → end
Full standardization	$(u - \mu_d)/s_d$	2.51	$0.998 \to 0.998$
Channel rescaling	u/s_d	2.70	$1.03 \to 1.94$
Scalar rescaling	u/s_{all}	2.64	$0.64 \to 0.08$
Centring	$u - \mu_d$	4.36	$0.64 \to 0.01$
No transformation	u	5.89	—

Table 9: **Alignment depth.** One-based block indices, JiT-L/16, 600 epochs, online targets and Conv + LN + TS. ΔFID is relative to $4 \to 18$; positive differences indicate higher FID.

Student → teacher block	FID↓	ΔFID
$4 \to 18$ (SiamFlow)	2.13	0.00
Student depth, teacher fixed at 18		
$2 \to 18$	2.26	+0.13
$6 \to 18$	2.18	+0.05
$8 \to 18$	2.27	+0.14
$12 \to 18$	2.39	+0.26
Teacher depth, student fixed at 4		
$4 \to 12$	2.25	+0.12
$4 \to 24$	2.28	+0.15
$4 \to 4$ (same-layer control)	2.91	+0.78

D Does cross-noise image retrieval predict generation quality?

Question. This section gives the retrieval protocol and controls for Section 6.3, which compares cross-noise image identification with generation FID.

What Recall@1 measures. For each of 1,024 validation images, we create two independently noised views at different noise levels. One view is a query; the other is placed in a candidate set containing one view of every image. We rank the candidates by cosine similarity of their mean-pooled, L2-normalized backbone features. A query succeeds if its top-ranked candidate comes from the same original image. Recall@1 is the percentage of successful queries, so higher values indicate better recognition of image identity across noise. It is distinct from the precision/recall metric used to evaluate generated-image distributions.

We use noise levels $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and average over the 20 ordered pairs of distinct levels. Baseline JiT and the online- and EMA-trained models all use sampling-EMA weights and null class conditioning. We compare retrieval and FID at the same checkpoints, saved every 100 epochs from 100 to 500. The diagnostic uses checkpoints from a separate online training run with the SiamFlow configuration; these are not the run used to report the main 2.13 result.

Result. Better cross-noise retrieval does not imply better generation. At every measured checkpoint, the EMA-trained model has the highest block-4 Recall@1, while the online-trained model has the lowest FID (Figure 7). At epoch 500, online and EMA obtain Recall@1 of 41.8% and 57.6%, respectively, but FID of 2.30 and 2.37. Across all four depths and five checkpoints, EMA has higher Recall@1 in $19/20$ comparisons under both the original and mean-subtracted protocols. The exception is block 12 at checkpoint 100: online versus EMA is 43.72% versus 43.39% without mean subtraction, and 71.18% versus 70.53% with it.

Evaluation protocol and checks. FID uses 50,000 generated samples, CFG 2.4, and 50 Heun steps at each saved checkpoint. These fixed-checkpoint values are distinct from the minimum-FID comparisons in the main text. Table 10 gives all points in Figure 7. To check whether the result depends on feature extraction, we also subtract the mean feature at each noise level before normalization or use true class labels in place of null conditioning. At checkpoint 500, EMA retains higher Recall@1 than online at all four measured depths under both variants (Table 11).

Table 10: **Cross-noise retrieval and generation across training.** Baseline JiT (B), online (O), and EMA-trained (E), all evaluated with sampling-EMA weights. Recall@1 (%) is averaged over 20 noise-level pairs. FID uses CFG 2.4 and 50,000 samples at the same checkpoints.

	FID↓			Recall@1, block 4			Recall@1, block 4, mean-sub.		
Checkpoint	B	O	E	B	O	E	B	O	E
100	4.15	**3.76**	4.10	10.0	47.5	**59.2**	59.4	71.7	**77.4**
200	2.95	**2.58**	2.78	13.1	49.2	**62.4**	58.4	69.5	**78.3**
300	2.70	**2.35**	2.56	13.7	46.0	**61.9**	56.0	68.3	**79.2**
400	2.52	**2.26**	2.51	10.9	44.9	**59.9**	53.8	68.4	**79.1**
500	2.48	**2.30**	2.37	7.3	41.8	**57.6**	51.9	69.6	**79.0**

Table 11: **Cross-noise retrieval under three extraction protocols.** Recall@1 (%) averaged over the 20 ordered pairs of distinct noise levels, epoch-500 checkpoints with sampling EMA weights, for Baseline JiT (B), online (O), and EMA-trained (E) models. The reported protocol uses the null class token. Mean-subtracted removes, at each noise level, the mean feature of the query set and of the candidate set before normalization. Class-conditioned uses the true label for every model.

	Reported			Mean-subtracted			Class-conditioned		
Block	B	O	E	B	O	E	B	O	E
4	7.3	41.8	57.6	51.9	69.6	79.0	8.1	47.9	65.2
12	13.3	27.4	30.4	59.8	63.5	67.3	17.4	35.5	36.9
18	35.1	35.8	46.1	69.6	63.8	73.9	45.3	45.1	54.5
24	17.9	28.4	44.7	58.0	60.3	67.4	19.7	33.5	46.7

E A CONDITIONAL-EXPECTATION VIEW OF THE TWO OBJECTIVES

This section proves the characterization in Section 3.1 and gives the details behind its uses. We write $z_t = tx_0 + (1-t)\epsilon$ with large t cleaner, fix the timesteps $t_s < t_t$ and the class condition c, and write $\mathcal{C} = (z_s, t_s, t_t, c)$ for the information available to the student, with $z_s = z_{t_s}$. The noise used to construct the teacher view z_{t_t} is independent of the student's noise and of x_0.

Flow matching. The loss is $w(t_s)\,\|F(z_s, t_s, c) - x_0\|^2$ with $w(t_s) = \max(1 - t_s, t_{\min})^{-2}$, a constant once t_s is fixed, so the pointwise minimizer is $F^\star = \mathbb{E}[x_0 \mid z_s, t_s, c]$ and the optimal velocity is $(F^\star - z_s)/\max(1 - t_s, t_{\min})$.

Alignment. Fix the target map ϕ, which comprises the target backbone, the projector, and token standardization, and let $U = \phi(z_{t_t})/\|\phi(z_{t_t})\|$ for one token. For an idealized predictor whose output q can be any function of $\mathcal{C}$,

$$\mathbb{E}\big[-\cos(q, U) \mid \mathcal{C}\big] = -\hat{q}^\top m, \quad m = \mathbb{E}[U \mid \mathcal{C}], \qquad \hat{q}^\star = \frac{m}{\|m\|} \quad \text{if } m \neq 0, \tag{4}$$

since $\hat{q}^\top m$ is maximized over unit vectors by the direction of m. Given x_0, t_t, and c, the teacher view does not depend on the student's noise, so z_{t_t} and z_s are conditionally independent given (x_0, t_s, t_t, c) and the tower property gives $m = \mathbb{E}[\bar{U}(x_0) \mid \mathcal{C}]$ with $\bar{U}(x_0) = \mathbb{E}[U \mid x_0, t_t, c]$. The optimum is therefore the direction of the teacher's normalized feature averaged over the teacher's noise and then conditioned on the student's view. If the conditional target direction is non-degenerate, $\|m\| < 1$ and the optimal mean cosine is below one. With a shared noise draw the two views are not conditionally independent given x_0, and the decomposition does not apply.

What the characterization establishes. Flow matching predicts a conditional image mean, whereas cosine alignment predicts the direction of a conditional feature mean. These mathematical targets differ, but their form alone does not compare generation quality. The retrieval experiment in Appendix D addresses a separate empirical question using mean-pooled backbone features.

Scope of the characterization. The derivation fixes the target map and assumes independently noised views and a predictor that can use all of $\mathcal{C}$. In the implemented model, the predictor receives only p_s, t_s, t_t, and the target map evolves during training. The result distinguishes the two prediction objectives; generation performance is assessed by the experiments.

How much a target change alters the gradient. For fixed q and a unit target u at angle $\varphi \in (0, \pi)$ from q, so that $\sin\varphi > 0$, the gradient of $-\cos(q, u)$ with respect to q is $g(u) = -Pu/\|q\|$ with $P = I - \hat{q}\hat{q}^\top$ (Equation (6)), and $\|g(u)\| = \sin\varphi/\|q\|$. For a second unit target u' at angle δ from u,

$$\frac{\|g(u') - g(u)\|}{\|g(u)\|} = \frac{\|P(u' - u)\|}{\sin\varphi} \leq \frac{\|u' - u\|}{\sin\varphi} = \frac{2\sin(\delta/2)}{\sin\varphi} \approx \frac{\delta}{\sin\varphi}, \tag{5}$$

The inequality is an equality when $u' - u$ is orthogonal to $\hat{q}$; the final approximation assumes small δ in radians. Table 12 measures each term for SiamFlow with an EMA teacher where the alignment loss acts, after the projector and token standardization. The inputs are 128 ImageNet validation images with four draws of timestep pairs and noise as in training, at the training weights of each checkpoint, and every quantity in a row is computed on the same inputs. At every checkpoint, and on all ten fixed (t_s, t_t) pairs from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ as well, the exact change of the gradient at the prediction is within 0.002 of $\delta/\sin\varphi$: the difference of the two targets is almost orthogonal to the prediction, and the per-token bound is nearly tight in these measurements. Over the tokens of a minibatch the change is 0.14–0.19; this quantity is bounded by the largest per-token value of $\delta/\sin\varphi$. Through the Jacobians of blocks 1–4 the change is 0.16–0.35, and single minibatches of 32 images range from 0.10 to 0.48. The bound does not cover these parameter-gradient values, which also depend on the Jacobian of the network. On the fixed pairs, φ falls from 28–42° at $t_s = 0.1$ to 9–14° for $t_s = 0.7$, $t_t = 0.9$, where the change at the prediction reaches 0.25–0.40.

Table 12: **Target rotation and gradient change on the same inputs.** SiamFlow with an EMA teacher; ranges over four draws of timestep pairs as in training. δ: per-token angle between the online and the EMA target after the projector and token standardization; φ: angle between the prediction and the EMA target; bound: per-token mean of $\delta/\sin\varphi$ (Equation (5)), an upper bound for the token column; output: exact relative change of the gradient with respect to the prediction, per token and over a minibatch; blocks 1–4: relative change of the alignment gradient with respect to the block parameters, median over 16 minibatches, range over the four blocks.

				Output		
Epoch	δ	φ	Bound	Token	Minibatch	Blocks 1–4
100	3.2–3.4°	16.7–18.7°	0.23–0.24	0.23–0.24	0.18–0.19	0.23–0.35
200	3.1–3.2°	18.4–20.5°	0.19–0.21	0.19–0.21	0.16–0.18	0.18–0.25
300	3.1–3.2°	19.6–21.8°	0.18–0.20	0.18–0.20	0.15–0.17	0.16–0.24
400	3.1–3.3°	21.4–23.7°	0.17–0.18	0.17–0.18	0.14–0.16	0.16–0.22
500	3.5–3.6°	23.9–26.4°	0.16–0.18	0.16–0.18	0.14–0.16	0.18–0.24

Testing a shorter averaging window. To support Section 6.2, we repeat the same paired measurements on the run with teacher EMA decay 0.99. The target-angle range falls from 3.1–3.6° to 1.1–1.9°, and the relative parameter-gradient change falls from 0.16–0.35 to 0.07–0.19. These ranges summarize checkpoints 100–500 using the protocol above. Despite smaller target and gradient differences, FID is 2.29 versus 2.13 for online alignment (Table 6). Faster averaging reduces the measured differences without matching online generation quality.

F Feature rescaling and the variance penalty

This section explains two properties used in Sections 5.1 and 5.2: how feature rescaling changes the predictor input, and why exact token standardization makes the variance penalty constant. The cosine-gradient identity also supplies the starting point for the target-sensitivity bound in Appendix E.

For nonzero q and p, negative cosine similarity has gradient

$$\nabla_q[-\cos(q, p)] = -\frac{1}{\|q\|}\left(\hat{p} - \cos(q, p)\hat{q}\right), \qquad \|\nabla_q[-\cos(q, p)]\| = \frac{|\sin\theta|}{\|q\|}. \tag{6}$$

This identity concerns the predictor output. The gradient reaching backbone parameters additionally contains the Jacobians of the predictor, token standardization, projector, and backbone.

Local scaling of the projector output. Consider one channel vector $u \in \mathbb{R}^N$ immediately before token standardization, $C = I - \mathbf{1}\mathbf{1}^\top/N$, and $s = \sqrt{u^\top C u/(N-1)}$. For $s > 10^{-4}$, exact arithmetic gives

$$S(u) = Cu/s, \qquad J_S(u) = \frac{C}{s} - \frac{(Cu)(Cu)^\top}{(N-1)s^3}. \tag{7}$$

For $c > 0$ with both standard deviations above the clamp, $S(cu) = S(u)$ and $J_S(cu) = J_S(u)/c$. Thus, rescaling removes a positive uniform change in the amplitude of the features supplied to the predictor, while centering alone retains that change. Because the GELU predictor is not scale-invariant, this is a possible explanation for the rescaling benefit in Section 5.2. The identities apply to the feature tensor with inactive clamps; they do not assert invariance of the full optimization process.

Variance regularization after token standardization. The implementation standardizes each image and channel using sample variance over N tokens, then computes the variance penalty over BN tokens on each process. With inactive clamps and exact arithmetic, each standardized channel has per-image sum zero and sum of squares $N-1$. The pooled sample variance is therefore

$$\mathrm{Var}_{b,n}(p_d) = \frac{B(N-1)}{BN-1}, \qquad \mathcal{V}(p) = \max\left(0, 1 - \sqrt{\frac{B(N-1)}{BN-1} + 10^{-4}}\right). \tag{8}$$

It is constant in upstream parameters under these conditions. For $B = 128, N = 256$, the standard deviation inside the hinge is approximately 0.9981. Although the hinge is active, its upstream gradient is zero because the standardized variance is fixed. Active clamps and finite-precision arithmetic can depart from this exact calculation. The FID values 2.13 with the penalty and 2.15 without it do not measure its gradient contribution (Table 4).

G Evaluation across model sizes and resolutions

The comparisons in Section 4.3 test whether the pixel-space alignment gain extends to a larger model and higher resolution. The XL/16 pair uses layers $5 \to 21$ and guidance 2.2; FID improves from 2.07 to 1.96. At 512×512, the L/32 pair uses guidance 2.5 and improves from 2.67 to 2.46 (Table 13). Both comparisons use 600 epochs and hold the sampling settings fixed within each pair.

Table 13: **ImageNet** 512×512**.** 600 epochs; L/32 rows differ only in alignment.

Model	Alignment	FID↓	IS↑
Baseline JiT-B/32	none	4.13	269.0
Baseline JiT-L/32	none	2.67	299.1
SiamFlow-L/32	online teacher	2.46	309.1

Checking the guidance choice. A final-checkpoint sweep tests whether the 512×512 improvement depends on a particular guidance setting. At guidance 2.2, 2.5, and 2.8, Baseline L/32 gives FID 2.81, 2.68, and 2.74, while SiamFlow gives 2.58, 2.47, and 2.55. SiamFlow improves FID at all three settings, and both models attain their best value at the shared guidance 2.5 in the tested grid $\{1.5, 1.9, 2.2, 2.5, 2.8, 3.2\}$. The sweep uses identical sampling noise at each guidance value within a model.

Applying the same pixel-space alignment settings to SD-VAE latent inputs did not improve FID; the effectiveness claims above concern the tested pixel-space models.