

Activation-Norm Maximization to Accelerate Training in Flow-Matching Transformers

Yash Belhe Wesley Chang Tzu-Mao Li Ravi Ramamoorthi Michaël Gharbi
University of California, San Diego Reve

Abstract

Flow-matching diffusion transformers are highly sensitive to their early optimization dynamics, where poor initialization often leads to slow convergence and sub-optimal sample quality. We propose a new data-dependent initialization technique that jointly minimizes the flow-matching loss and a lightweight regularizer that maximizes activation norms for a short time at the beginning of training (less than 0.1% of the total training steps). Our method leads to faster loss reduction and lower final FID. The regularizer is just a few lines of code, has minimal overhead, and can be seamlessly integrated into existing pipelines. At 400k steps, our method significantly reduces the FID for the S/2 (-4.62%), B/2 (-13.21%), L/2 (-21.95%) and XL/2 (-34.34%) SiT variants. For the largest XL/2 model, even after 2 million steps (2M), FID is reduced by 24%, by simply using our initialization for the first 2k steps and continuing with regular flow-matching training thereafter. In fact, we match the baseline SiT’s final FID at 7M steps in just 1.2M steps, a 5.8× speedup. Our method can also be used in conjunction with the recent representation alignment (REPA) regularization, further improving upon it, and outperforms a recently proposed dispersion regularization while requiring lesser overhead than it.

1. Introduction

Diffusion transformers have recently achieved strong generative performance [6, 7, 16, 17], but their training dynamics remain sensitive to early optimization. Poor initial trajectories often slow convergence or lead to sub-optimal quality despite large-scale training [17].

We introduce a lightweight, plug-and-play activation-norm-maximizing regularizer that is only applied briefly at the beginning of training Fig. 1. During this initial phase, the model is optimized with both the standard flow-matching objective and our regularizer. After this warm-up phase, we continue training with the regular flow-matching

```
def reg_loss(F, alpha):  
    # F: [B, L, D] (batch, layers, flattened  
    # tokens*hidden), flattened output of all  
    # transformer blocks.  
    s = sum(F * F, dim=2) # [B, L]  
    e = exp(-s) # [B, L]  
    g = 1 + (alpha / L) * e.sum(dim=1) # [B]  
    return log(g).mean()
```

Figure 1. Our activation norm maximization loss Eq. (2) can be implemented easily in code.

objective only. We view this procedure as a data-dependent warm-up that increases the energy of internal representations. Intuitively, it amplifies the magnitude of features already favored by the flow-matching loss. We find that this initialization yields more favorable training trajectories for flow-matching models, leading to better FID in converged models and faster convergence overall.

Despite its simplicity, this initialization yields consistent and lasting gains. Across model sizes (S/2, B/2, L/2, XL/2), we observe lower converged FID [10] than a strong SiT baseline, with increasingly larger relative improvements for larger models (e.g., 34.34% at XL/2). The warm-up is compute- and memory-efficient, requires no additional data or infrastructure, and integrates into existing diffusion pipelines with a few lines of code. Its effect persists even after millions of iterations, and remains robust when the warm-up dataset differs from the training dataset (e.g., Places365 [25] → ImageNet [5]).

Our analyses show why and when the warm-up helps: (i) duration: brief schedules ($\approx 2k$ steps) outperform longer ones; (ii) interaction with the flow-matching loss: the warm-up provides no benefit without the flow-matching loss, indicating it reinforces meaningful features rather than indiscriminately inflating activations; (iii) sensitivity to conditioning: benefits vanish when shuffling x_t or v^* , implicating correlations central to the flow objective; (iv) placement: applying the regularization to the block output (after subtracting the residual) and across all layers performs best;

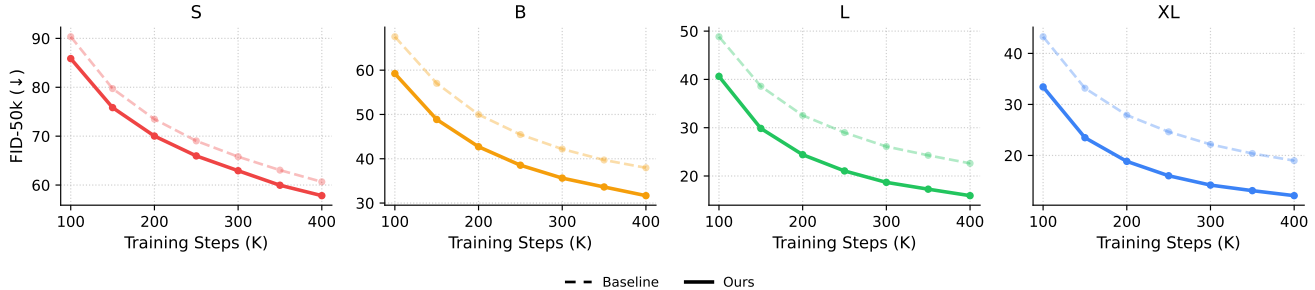


Figure 2. **Activation norm maximization improves FID for all model sizes.** Our initialization outperforms SiT at all model sizes (S/2, B/2, L/2, XL/2). Importantly, its relative improvement is stronger for larger models, making it a cheap and effective regularizer for large-scale runs.

and (v) metric variants: an ℓ_2^2 formulation outperforms ℓ_1 and ℓ_3^3 alternatives.

In summary, we view activation-norm maximization as a practical, data-driven initialization for diffusion transformers that:

- Consistently improves the final FID of the converged model, and its convergence rate, across a variety of model sizes and training regimes.
- Has minimal overhead, because it is applied for less than 0.1% of the total training time and requires no additional forward passes or cross-sample synchronization.
- Is Plug-and-play. It involves no external model, no additional data, and no inter-accelerator communication, or memory-heavy loss computation.
- Is robust for a wide range of hyperparameters.
- Is complementary to other regularizers and compatible with prior methods (e.g., REPA [23]).

Our proposed warm-up method is a straightforward, drop-in addition to any diffusion training pipeline, that improves the convergence rate and final model quality of flow-matching transformer models.

2. Related Work and Background

Diffusion models and flow matching. Diffusion models [11, 19, 20, 20], and more recently Flow matching [1, 2, 13–15], which we focus on, are state-of-the-art generative models that synthesize images starting from a Gaussian noise map. Flow matching trains a time-conditioned velocity field $v_\theta(x_t, t, y)$ to match a target velocity v^* along a simple path between a data x and a prior sample $\epsilon \sim \mathcal{N}(0, I)$. We use the straight-line path $x_t = (1 - t)x + t\epsilon$ with $t \sim \mathcal{U}(0, 1)$, whose target velocity is $v^* = \epsilon - x$. The flow is also conditioned on an extra variable, y , which could be an object class or a text prompt in text-to-image applications. The standard flow-matching objective is

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{x, y, \epsilon, t} \left[\left\| v_\theta(x_t, t, y) - v^* \right\|_2^2 \right]. \quad (1)$$

In this paper, v_θ denotes a diffusion transformer trained with flow matching in the latent space z of a pre-trained Variational Autoencoder (VAE) [18].

Initializing diffusion transformers. Initialization is important for deep networks [8, 9, 24]. In residual architectures, zero-initializing the residual branch so that each block starts as an identity mapping stabilizes very deep models and improves training dynamics [8, 9, 24]. Diffusion Transformers (DiT) [17] and its follow-up Scalable interpolant Transformers (SiT) [16] adopt an analogous idea via AdaLN-Zero—initializing the adaptive LayerNorm scale to zero—so blocks begin near-identity which improves convergence and FID. With AdaLN-Zero, the residual branch initially produces zero output; the block output minus the input is zero at initialization. We start from this initialization and improve upon it with our activation-norm maximization.

Regularizing diffusion transformers. Recent work has shown that regularization of the internal activations of diffusion transformers can significantly improve generation quality [12, 22, 23]. Yu et al. [23] show that aligning the internal activations of a diffusion transformer with a pre-trained vision foundation model helps. However, at each training step this requires computing features for input images. Training the vision foundation model itself requires significant compute as well as larger datasets. Not only does our method not require additional data and only requires minimal compute, but it is complementary to Yu et al.’s method and helps accelerate their training as well. Wang et al. [22] propose a contrastive learning [4] based regularizer that maximally separates activations for a batch of input images. While it does not require additional data, scaling contrastive learning to larger batch sizes requires significant inter-accelerator communication and complex algorithms to minimize overhead [3]. Later, we discuss how our method relates to and improves upon theirs.

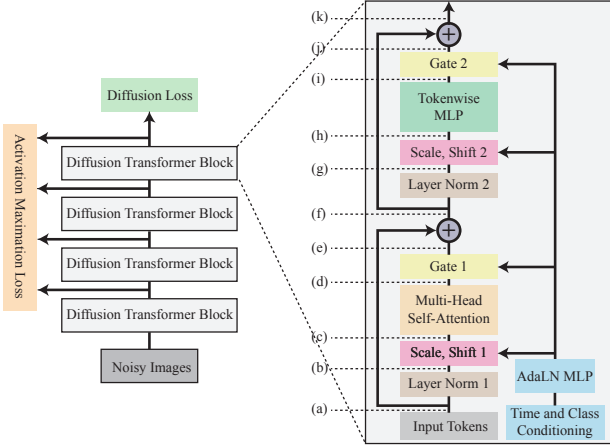


Figure 3. Our regularization maximizes the activations collected at the outputs of each transformer block (left). Right: There are several choices for which activations to maximize within each block, labeled (a) to (k). We find that applying the regularization to the output (k) minus the input (a) of each block works best.

3. Activation-Norm Maximization

Overview. We introduce a lightweight regularization strategy that encourages higher activation magnitudes during the early stages of flow-matching training and demonstrate benefits on the popular SiT [16] family of transformers. This regularization is applied only during a short warm-up period. It can be interpreted as a data-dependent initialization procedure. Despite its simplicity, our method yields consistent improvements in both convergence speed and generative quality across model sizes.

Method. Our regularization loss maximizes the norms of the activations of all transformer blocks $F_j(x_t, t, y)$, where $j \in \{1, \dots, L\}$ is the block index, see Fig. 3. For brevity, we shorten $F_j(x_t, t, y)$ to F_j in the following text, all losses are per-input tuple (x_t, t, y) and averaged over the batch. Let $T \in \mathbb{N}^*$ denote the number of tokens and $H \in \mathbb{N}^*$ the hidden size. For a single input, the post-block activation at layer j is in $\mathbb{R}^{T \times H}$; we flatten it to $F_j \in \mathbb{R}^{TH}$ and take $\|F_j\|_2$ over all elements. Unless otherwise stated, we take F_j to be the block output after subtracting the residual input (placement (k) in Fig. 3). The loss has the form of log-sum-exp and is given by:

$$\mathcal{L}_{\text{reg}} = \log \left(1 + \frac{\alpha}{L} \sum_{j=1}^L \exp(-\|F_j\|_2^2) \right), \quad (2)$$

where $\alpha > 0$ is a scalar hyperparameter that controls the target activation scale. Since both log and exp are monotonically increasing functions, the loss penalizes small activation norms and encourages them to grow larger. It attains its

minimum value, when the exponential $\exp(-\|F_j\|_2^2) \approx 0$ saturates due to the norms $\|F_j\|_2^2$ growing sufficiently large. Its derivative with F_k reveals further insights¹

$$\frac{\partial \mathcal{L}_{\text{reg}}}{\partial F_k} = -2 \underbrace{\frac{\frac{\alpha}{L} \exp(-\|F_k\|_2^2)}{1 + \frac{\alpha}{L} \sum_{j=1}^L \exp(-\|F_j\|_2^2)}}_{\text{softmax-like}} F_k. \quad (3)$$

The gradient is composed of two terms: a softmax-like scaling term and the activation vector F_k itself. The second term encourages individual components of F_k to be large, while the first term scales down the entire gradient as the norm of F_k increases, approaching zero when the norm of F_k becomes sufficiently large. This regularization thus performs a constrained maximization of the activation norms.

During the warm-up (first 2,000 iterations; $< 0.1\%$ of training), the model is optimized jointly with the flow-matching loss \mathcal{L}_{fm} :

$$\mathcal{L} = \mathcal{L}_{\text{fm}} + \lambda \mathcal{L}_{\text{reg}} \quad (4)$$

where $\lambda > 0$ controls the strength of the regularization. After the warm-up, training proceeds normally using only \mathcal{L}_{fm} . Because the regularization is applied for such a short phase, its effect is best viewed as initializing the network to a more favorable region of weight space.

What happens during the warm-up? At initialization, the activations F_j are all zero (due to AdaLN-Zero), so the gradient of the regularization loss $\frac{\partial \mathcal{L}_{\text{reg}}}{\partial F_k}$ is zero. However, after the first optimization iteration, the flow-matching loss \mathcal{L}_{fm} makes F_k non-zero, adjusting its components F_{ki} to maximally reduce the flow-matching loss. Immediately, the regularization loss *boosts* the non-zero activation components F_{ki} selected by the flow-matching loss. The maximization for each layer k continues until its corresponding exponential $\exp(-\|F_k\|_2^2)$ saturates. Although this maximization is only applied during the short warm-up, as we show in the following section, its effects persist even after millions of iterations, resulting in significantly lower flow-matching loss and FID (see Sec. 4 and Tab. 1).

4. Analysis and results

We proposed a regularization technique which encourages higher ℓ_2 norms of internal activations during a warm-up stage and hypothesized that this regularization boosts features selected by the flow-matching loss. Our method has very minimal computational and memory overhead, it requires no additional data and can be easily integrated into existing diffusion pipelines with just a few lines of code,

¹We have made the simplifying assumption that F_k is not a function of F_j for $k \neq j$ for notational simplicity.

see Fig. 1. In this section, we validate the effectiveness of our method, ablate its design choices and show evidence to support our hypothesis.

Setup. We follow the exact SiT (Scalable Interpolant Transformer) [16] implementation, which is our baseline. In all experiments, our model uses the same number of total iterations as the baseline including its warm-up. All experiments perform flow-matching in the latent space of the SD-VAE (Variational Autoencoder; VAE) [18] on ImageNet-256 [5] unless noted otherwise. We perform ablations on the B/4 at 200k iterations and additionally include results for XL/2 at 100k iterations for some experiments. FID is computed using an ODE sampler with 250 sampling steps for all results unless noted otherwise. For more experimental details please see the supplemental.

For all experiments, we use $\lambda = 10$ and $\alpha = 70$; we ablate their effects in the supplemental.

Our method provides consistent improvements across all model sizes. We compare our method with the baseline SiT across models of different sizes (S/2, B/2, L/2 and XL/2) in Fig. 4. At 400k steps, we find that our method has a lower FID than the baseline for all model sizes (-4.62%, -13.21%, -21.95%, -34.34%) respectively. Even at 2M steps, our XL/2 model has an FID of 24.23% lower than the baseline, indicating that our initialization benefits persist deep into model training; see visual results in Fig. 2. In fact, we only require 1.2M steps to match the final FID of XL/2 computed at 7M steps, a $5.8\times$ speedup. It also has a lower flow-matching loss than the baseline at all model sizes across all timesteps, indicating across-the-board improvements (results in supplement). The relative FID improvements for our method are larger for larger models, which makes it especially useful in these settings. Please refer to the supplement for more comparisons with the baseline.

Warm-up duration. Keeping the total number of iterations constant, we vary the duration of the warm-up ($\mathcal{L} = \mathcal{L}_{\text{fm}} + \lambda\mathcal{L}_{\text{reg}}$) and flow-matching training ($\mathcal{L} = \mathcal{L}_{\text{fm}}$). We find that a brief warm-up (2,000 iterations) is sufficient to achieve all the benefits of our method for both large (XL/2) and small (B/4) models. For larger models, we notice that a very long warm-up (100k steps) can somewhat degrade performance. However, even this setting significantly outperforms the baseline, see Tab. 2.

Interaction with the flow-matching loss. We hypothesized that our regularization boosts features selected by the flow-matching loss. To test this, we remove the flow-matching loss during the warm-up and train only with \mathcal{L}_{reg} . Naively doing so would result in the regularizer having no

Table 1. Our method significantly outperforms the baseline SiT [16] across all model sizes; relative improvements increase for larger models. With classifier-free guidance, our method, trained for 1.5M fewer iterations than the baseline, achieves 18.29% lower FID. All results use ODE sampling with 250 steps.

Run	Steps	FID↓
SiT-S/2 (Baseline)	400k	60.63
SiT-S/2 (Ours)	400k	57.83 (-4.62%)
SiT-B/2 (Baseline)	400k	36.49
SiT-B/2 (Ours)	400k	31.67 (-13.21%)
SiT-L/2 (Baseline)	400k	20.41
SiT-L/2 (Ours)	400k	15.93 (-21.95%)
SiT-XL/2 (Baseline)	400k	18.46
SiT-XL/2 (Ours)	400k	12.12 (-34.34%)
SiT-XL/2 (Baseline)	1M	12.18
SiT-XL/2 (Ours)	1M	8.70 (-28.57%)
SiT-XL/2 (Baseline)	2M	10.11
SiT-XL/2 (Ours)	2M	7.66 (-24.23%)
SiT-XL/2 (Baseline) w/ cfg	4M	2.46
SiT-XL/2 (Ours) w/ cfg	2.5M	2.01 (-18.29%)

Table 2. **Effect of regularization duration.** We find that 2k steps is sufficient for our regularization for both model sizes; excessive regularization hurts FID, especially for the larger model.

Method	\mathcal{L}_{reg} steps	Training steps	FID↓
SiT-B/4 (Baseline)	-	200k	70.61
SiT-B/4 + \mathcal{L}_{reg} (Ours)	2k	200k	62.85
SiT-B/4 + \mathcal{L}_{reg}	100k	200k	62.39
SiT-XL/2 (Baseline)	-	100k	43.28
SiT-XL/2 + \mathcal{L}_{reg} (Ours)	2k	100k	33.43
SiT-XL/2 + \mathcal{L}_{reg}	100k	100k	37.56

effect since the activations F_i are initialized to zero. Instead, we initialize the weights using a normal distribution with a small standard deviation $\sigma > 0$. Without \mathcal{L}_{fm} , the model shows no benefit, indicating that random activation amplification alone is insufficient. The regularization thus enhances meaningful activation patterns already induced by the flow-matching loss, rather than merely increasing activation energy (see Tab. 3).

Sensitivity to conditioning variables. The flow-matching loss, $\mathcal{L}_{\text{fm}} = \|v_t(x_t, t, y) - v^*\|_2^2$, aligns the predicted velocity $v_t(x_t, t, y)$ with the velocity target $v^* = \epsilon - x$ given the noisy input $x_t = (1-t)x + t$, timestep t , and class label y . Including \mathcal{L}_{fm} in the warm-up is essential, but it is unclear what role each of the four conditioning or target variables (x_t, t, y, v^*) play in the regularization. To identify which conditioning signals the regularization



Figure 4. Selected samples on ImageNet-256 generated using our XL/2 model with $\text{cfg}=4.0$.

Table 3. Regularization \mathcal{L}_{reg} only helps when coupled with flow-matching \mathcal{L}_{fm} , which improves our FID over the baseline (b). Removing it and amplifying random activations is insufficient (c,d).

Method	Init. Phase \mathcal{L}	AdaLN Init.	FID \downarrow
SiT-B/4 (Baseline)	-	AdaLN-Zero	70.61
SiT-B/4 (Ours)	$\mathcal{L}_{\text{fm}} + \lambda\mathcal{L}_{\text{reg}}$	AdaLN-Zero	62.85
SiT-B/4 ($\sigma = 0.02$)	\mathcal{L}_{reg}	$\mathcal{N}(0, 0.02)$	70.68
SiT-B/4 ($\sigma = 0.0002$)	\mathcal{L}_{reg}	$\mathcal{N}(0, 0.0002)$	68.42

amplifies, we perturb each input component by shuffling it across the batch during the warm-up. Batch shuffling preserves the marginal distributions of all variables but breaks their cross-variable correlations. We observe that shuffling x_t or v^* removes all benefits, whereas shuffling t or y has little effect; see Tab. 4. This indicates that the correlations between x_t and v^* —as coupled through the flow-matching loss—are necessary for the regularizer to be effective.

Using a different dataset for the warm-up. The evidence above suggests our regularizer amplifies correlations between x_t and v^* that are induced by the flow-matching loss. A natural question is whether the dataset used during the warm-up must match the dataset used for subsequent training. If the regularizer primarily boosts generic natural-image features, then pre-initializing on a related dataset should offer similar benefits. We observe exactly this: ini-

Table 4. **What conditioning variables does regularization target?** Shuffling along x_t and v^* during the warm-up phase degrades our method’s performance, indicating that their coupling through the flow-matching loss is necessary for effective regularization; no-shuffling, shuffling t and shuffling y all perform near-best.

Model	Shuffle along	FID \downarrow
SiT-B/4 (Baseline)	-	70.61
SiT-B/4 (Ours)	-	62.85
SiT-B/4 (Ours)	x_t	70.36
SiT-B/4 (Ours)	v^*	69.78
SiT-B/4 (Ours)	t	63.18
SiT-B/4 (Ours)	y	62.75

tializing on Places365 [25] and then training on ImageNet yields improvements close to initializing and training on ImageNet; see Tab. 5. We use the same warm-up duration for both datasets (2,000 iterations). However, initializing using synthetic images only like sinusoidal wave patterns with random colors, frequencies and orientations does not work, showing that our initialization method non-trivially adapts to natural image statistics; see supplemental for sample images from the sinusoidal dataset.

What activations should be maximized? Next, we investigate where within a transformer block our regularization should be applied. Across several configurations and hyperparameter settings, we find that applying it to the

Table 5. **Different warm-up dataset.** Using a different (but related) dataset like Places365 [25] during the warmup-phase can also work well; warming up on less related data, like colorful sinusoidal patterns is insufficient.

Model	Iters	Init. Phase	Training	FID↓
SiT-B/4 (Baseline)	-	-	ImageNet	70.61
SiT-B/4 (Ours)	2k	ImageNet	ImageNet	62.85
SiT-B/4 (Ours)	2k	Places365	ImageNet	63.82
SiT-B/4 (Ours)	2k	Sinusoids	ImageNet	69.16

Table 6. **Best location to maximize norms.** We try maximizing the activation norms at several locations (b-k in Fig. 3); all subtract the input residual (a). Maximizing the activation norm of the layer output works best.

Model	Activation location	FID↓
B/4 (Baseline)	-	70.61
SiT-B/4 (Ours)	(b) after Layer Norm 1	339.76
SiT-B/4 (Ours)	(c) after Scale, Shift 1	69.02
SiT-B/4 (Ours)	(d) after Multi-Head Self-Attention	69.93
SiT-B/4 (Ours)	(e) after Gate 1	67.9
SiT-B/4 (Ours)	(f) after Middle Residual	66.14
SiT-B/4 (Ours)	(g) after Layer Norm 2	66.57
SiT-B/4 (Ours)	(h) after Scale, Shift 2	65.97
SiT-B/4 (Ours)	(i) after Tokenwise MLP	69.50
SiT-B/4 (Ours)	(j) after Gate 2	72.86
SiT-B/4 (Ours)	(k) after Final Residual	62.85

Table 7. **What layer to apply regularization to?** We find that it is most effective to apply our regularization to all layers instead of any single one. This suggest that, beyond simply increasing activation norms, our regularizer enables a data-dependent co-adaptation of internal features that leads to a better final trained model.

Placement	Layer idx	FID↓
SiT-B/4 (Baseline)	-	70.61
SiT-B/4 (Ours)	all	62.85
SiT-B/4 (Ours)	11	68.96
SiT-B/4 (Ours)	6	68.16
SiT-B/4 (Ours)	1	68.53

block output after subtracting the residual connection works best, see Tab. 6.

We also test applying the regularization only to certain layers and find that it consistently underperforms applying the regularization to all layers, as seen in Tab. 7.

Metric Variants We also experiment with different distances used inside the regularizer across $\|\cdot\|_2^2$ (squared ℓ_2), $\|\cdot\|_1$ (ℓ_1), and $\|\cdot\|_3^3$ (cubed ℓ_3). We sweep across three different regularization strengths $\lambda \in \{3.0, 10.0, 30.0\}$ and report the best result for each variant. Overall, $\|\cdot\|_2^2$ performs

Table 8. **Metric used in \mathcal{L}_{reg} .** Of the three choices below, we found ℓ_2^2 to work best; all three metrics outperform the baseline.

Model	Metric	FID↓
SiT-B/4 (Baseline)	-	70.61
SiT-B/4 (Ours)	ℓ_2^2	62.85
SiT-B/4 (Ours)	ℓ_1	63.60
SiT-B/4 (Ours)	ℓ_3^3	66.08

Table 9. **Flow matching in E2E-VAE’s [12] latent space and compatibility with REPA [23].** Our regularizer’s benefits are not limited to the SD-VAE’s [18] latent space (used in all other experiments), it also generalizes to the E2E-VAE’s latent space. It is also complementary to the state-of-the-art regularization technique REPA and further improves upon it.

Placement	Steps	FID↓
SiT-B/1 + E2E-VAE + REPA (Baseline)	400k	12.8
SiT-B/1 + E2E-VAE + REPA (Ours)	400k	10.94 (-14.5%)
SiT-B/1 + E2E-VAE + REPA (Baseline)	1.6M	8.67
SiT-B/1 + E2E-VAE + REPA (Ours)	1.6M	8.19 (-8.19%)
SiT-XL/1 + E2E-VAE + REPA (Baseline)	400k	3.46
SiT-XL/1 + E2E-VAE + REPA (Ours)	400k	2.83 (-18.2%)

best; see Tab. 8.

Compatibility with other regularizations and VAEs

Thus far, all experiments perform flow-matching in the latent space of the SD-VAE; our method’s benefits are not tied to this specific choice and generalize to the E2E-VAE [12] as well. In addition, our method can be directly combined with other regularization techniques such as Representation Alignment (REPA) [23]. Performing latent flow-matching with E2E-VAE and REPA regularization already significantly outperforms the baseline SiT [12] — applying our method further accelerates convergence Tab. 9.

Unconstrained maximization. An alternative to our constrained maximization in \mathcal{L}_{reg} (see Eq. (2)) is to maximize the norms of the activations of all transformer blocks F_j unconstrainedly as $\mathcal{L}_{\text{unconstrained}} = -\sum_{j=1}^L \|F_j\|_2^2$. Doing so results in the training diverging (FID > 300) as the norms grow unbounded. We found that our method is only mildly sensitive to the value of α , as we show in supplemental.

Connection to the dispersion (DaD) loss [22]. Our regularization in Eq. (2) is closely related to the InfoNCE-style

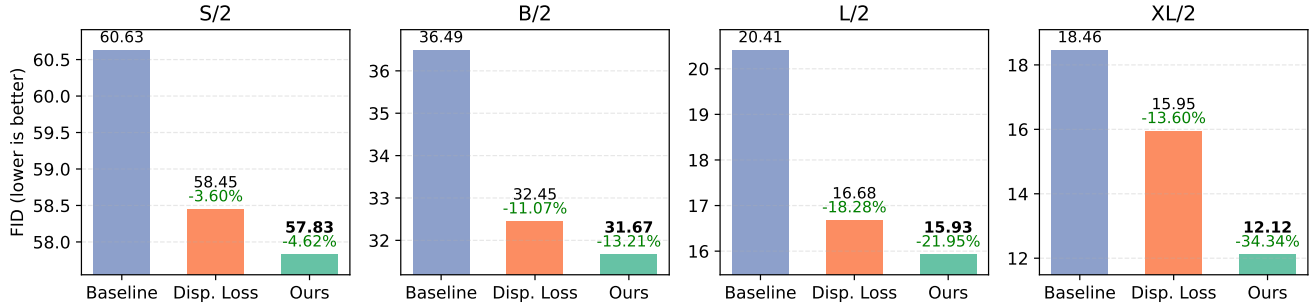


Figure 5. **Activation maximization outperforms the dispersion loss from [22] for all model sizes.** Despite only being only applied in the first 2k iterations, our loss \mathcal{L}_{reg} outperforms the dispersion loss applied continuously throughout training; see Sec. 4 for further discussion.

diffuse and disperse (DaD) loss [21, 22]:

$$\mathcal{L}_{\text{disp}} = \log \left(1 + \sum_{m \neq n} \exp(-\|F_k(x_m) - F_k(x_n)\|_2^2) \right), \quad (5)$$

which increases pairwise distances between activations $F_k(x_m)$ and $F_k(x_n)$ within a batch at layer k . We hypothesize that most of the benefit from the DaD loss comes from maximizing activation norms in the early optimization step: simply scaling up $F_k(x)$ reduces Eq. (5).

To test our hypothesis, we applied DaD only for the first 2,000 steps and then trained with flow matching thereafter. This short DaD warmup—likely too brief for effective contrastive representation learning, yet sufficient to increase activation norms (cf. Tab. 2)—achieved FID of 58.10 (S/2) and 33.18 (B/2), whereas applying DaD for the full 400k steps yielded 58.45 (S/2) and 32.45 (B/2). Since most of the improvement is realized within the first 2k steps, this suggests that activation-norm growth drives much of DaD’s benefit.

Our activation maximization is both superior (superior final FID) and cheaper (it avoids $O(B^2)$ pairwise computations and applies only at the start, where B is the batch size); see Fig. 5.

We can view our loss Eq. (2) as a variant of Eq. (5), with $F_k(x_n) = 0$, and where the sum is over layers rather than image pairs. Despite this formal similarity, the two objectives differ. Ours is derived from energy maximization, avoids $O(B^2)$ pairwise computations and cross-device communication, and only needs to be applied early in training.

Additional comparisons and ablations. We provide additional experiments, including the effect of varying the hyperparameters α and λ in the supplemental.

5. Conclusion and future work

We introduce a simple, optimization-based, initialization technique that strictly improves diffusion flow matching models and their convergence rate. Our method is effective, leading to a significant reduction in FID across model configurations. It can easily be integrated in existing pipelines in a few line of codes, and a few thousand optimization steps at the beginning of training. It requires no external data or expansive pretraining of external features. Our initialization is also efficient: it requires no extra forward pass through the model beyond the flow-matching loss.

Although we have not found definitive compelling evidence for why maximizing activation norms leads to better models, we hypothesize the maximization procedure breaks symmetry early and favors more discrete decision patterns in the model’s attention operators. Future work may study the mechanism behind our method’s benefits to perhaps derive an optimization-free network initialization scheme based on it. Understanding if these gains translate to text-to-image models, and also other transformers beyond diffusion (e.g., language models) would be interesting. In the meantime, activation maximization is a virtually cost-free technique to boost any flow-matching transformer!

Acknowledgements. This work was supported in part by NSF grant 2341952, NSF Chase-CI grants 2100237 and 2120019, gifts from Adobe, Google and Qualcomm, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing. We also thank Reve AI for their gracious support with compute.

References

- [1] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023.
- [2] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2025.

- [3] Zesen Cheng, Hang Zhang, Kehan Li, Sicong Leng, Zhiqiang Hu, Fei Wu, Deli Zhao, Xin Li, and Lidong Bing. Breaking the memory barrier: Near infinite batch size scaling for contrastive loss. 2024.
- [4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 539–546 vol. 1, 2005.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [6] 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, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.
- [7] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer, 2024.
- [8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. 2018.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.
- [10] 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, 2018.
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.
- [12] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025.
- [13] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.
- [14] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024.
- [15] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.
- [16] 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. 2024.
- [17] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2022.
- [18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [19] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.
- [20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.
- [21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.
- [22] Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization, 2025.
- [23] 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*, 2025.
- [24] Hongyi Zhang, Yann Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *ArXiv*, abs/1901.09321, 2019.
- [25] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.