ON THE SPECTRAL PROPERTIES OF GENERATIVE DENOISER JACOBIANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative denoising models, such as diffusion and flow-matching, learn to sample from complex distributions by training a deep neural network denoiser to recover clean data from noise-corrupted samples. While such models are typically compared on the quality of their synthesized samples, these metrics provide limited insight into how the underlying denoiser, which drives generation, differs. In this work, we propose to analyze the spectrum of the denoiser Jacobian as a tool to characterize these differences. Across pre-trained denoising models, we observe that better generative performance is associated with larger Jacobian eigenvalues. Motivated by this, we introduce a regularization scheme that controls the Jacobian spectrum by training the denoiser on perturbed inputs, with perturbations suppressing or amplifying Jacobian responses. On ImageNet, we test whether directly modifying the Jacobian spectral properties leads to improved generations. Our findings suggest that denoisers benefit from both strengthening responses along data-relevant principal eigen-directions and suppressing the noisy, data-irrelevant ones. This establishes the denoiser Jacobian as a useful tool for identifying differences between generative denoising models.

1 INTRODUCTION

Denoising generative models, such as diffusion (Ho et al., 2020) and flow-matching (Lipman et al., 2023), learn to sample from complex distributions using a deep neural network denoiser. For natural images, the denoiser looks at noise-corrupted samples and utilizes patterns learned from the training data to predict the underlying clean image. By repeatedly making such predictions, denoising models gradually transform random Gaussian noise into realistic images (Song et al., 2021a).

Once trained, the typical approach for comparing models is to evaluate the quality of their generations. While the literature has developed a multitude of metrics (Heusel et al., 2017; Sajjadi et al., 2018), these ultimately only characterize differences in model outputs, therefore providing limited insight into how the learned denoising function differs, e.g., the patterns used in denoising images. The question then is *why* one denoiser synthesizes samples better than the other, and since sample quality alone cannot answer it, we turn to the properties of the denoising function itself.

In this work, we examine the Jacobian of the learned denoiser as a means to probe its internal properties and reveal differences across models. The Jacobian specifies how small changes in the noisy input map to changes in the denoised output, which inherently captures the dependencies learned by the trained model (Graikos et al., 2026; Lukoianov et al., 2026). For instance, when perturbing a single noisy input pixel, the Jacobian reveals which pixels in the output are affected and in what way, encoding the covariances used to effectively denoise samples.

Since the Jacobian acts as a learned covariance matrix, we turn to spectral analysis, i.e., eigenvectors and eigenvalues, to describe its behavior. Starting from pre-trained generative models (SiTs (Ma et al., 2024)), we uncover an intriguing relationship: models that are better in generative metrics also exhibit larger denoiser Jacobian eigenvalues, which correspond to sharper responses of the learned denoising function along the eigen-directions.

Motivated by this observation, we next ask whether the reverse also holds: does altering the Jacobian spectrum change generation quality? We first introduce a simple regularization scheme that controls the Jacobian spectrum by perturbing the denoiser input during training. We show that, with

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.

an appropriate choice of perturbation, we can either suppress spurious directions in the Jacobian spectrum or encourage larger gain along principal eigenvectors, without ever explicitly computing the eigendecomposition during training.

Finally, we apply this regularization to ImageNet models, a setting comparable to the pre-trained model analysis. Our findings show that spectral regularization improves generative metrics both when encouraging large responses along eigen-directions and when suppressing noisy, data-irrelevant responses. We further show that these two objectives are composable, yielding cumulative improvements when applied together and suggesting that an effective model should exhibit both properties in its Jacobian spectrum.

By providing a framework that pinpoints differences between models through their Jacobian spectra, we target a more complete view of how training shapes the learned denoising function. With regularization becoming increasingly popular for training speed and generation quality (Yu et al., 2025), but yet partially understood (Singh et al., 2025), any principled effort to describe these differences is crucial, both for understanding existing methods and designing future objectives. Our results in this work suggest that the denoiser Jacobian is a useful tool for doing so.

We summarize our contributions as follows:

- We establish spectral analysis as a useful tool for revealing differences between generative denoising models. Specifically, we show that better-performing models exhibit larger eigenvalues in their denoiser Jacobian, uncovering a previously unknown relationship between Jacobian spectrum and generative performance.
- We propose a regularization scheme that controls the denoiser Jacobian by perturbing model inputs during training. For different perturbations, we show how we can selectively suppress or amplify the Jacobian response along random or the eigen-directions.
- We apply the proposed regularizer to ImageNet and demonstrate that generative performance improves when increasing principal Jacobian eigenvalues. We further show that this effect is complementary to suppressing the noisy, data-irrelevant responses, with the two yielding cumulative improvements when combined.

2 Related Work

Denoising generative models Denoising is the primary approach for learning to generate physical signals (e.g., images Rombach et al. (2022); Esser et al. (2024), audio (Evans et al., 2024)). Although the idea of modeling data distributions by denoising is not new (Bengio et al., 2013; Sohl-Dickstein et al., 2015), recent advances have produced capable generative models with both algorithmic (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023) and architectural improvements (Nichol & Dhariwal, 2021; Peebles & Xie, 2023; Ma et al., 2024). Regardless of specific formulation, every model is built using a learned denoiser, whose spectral properties are the focus of this work.

Denoiser Jacobians and spectral analysis. Recent work has looked at the Jacobian of the learned denoiser, which reveals dependencies between input and output pixels, to probe the learned statistics of the data distribution (Lukoianov et al., 2026; Graikos et al., 2026). Regarding spectral properties, Manor & Michaeli (2024) related the posterior uncertainty to the Jacobian in (non-generative) denoisers and employed a similar power iteration algorithm to compute principal components. Kadkhodaie et al. (2024) showed that eigenvectors of a generative denoiser form a data-adaptive harmonic basis. While these works motivate spectral analyses of the Jacobian, they mostly focus on individual models. In contrast, we draw comparisons across different denoisers, which only then reveals how spectrum relates to generation quality.

Training regularization Regularizing the training of diffusion and flow-matching models has recently garnered widespread attention, particularly with the success of Representation Alignment (REPA (Yu et al., 2025)). By showing that it accelerates training and benefits generation quality, regularization offers an appealing research direction for improving generative models. This has motivated works that regularize with a student-teacher framework (Chefer et al., 2026), or by utilizing contrastive objectives (Stoica et al., 2025; Wang & He, 2025).

Closer to our work, Ning et al. (2023) perturb denoiser inputs during training with Gaussian noise to mitigate the mismatch between inputs seen in training and sampling, while Scarvelis & Solomon

(2024) showed that perturbations can impose nuclear norm regularization on the Jacobian without explicitly constructing it. Interestingly, Alain & Bengio (2014) previously connected denoising and Jacobian regularization to the local geometry of the data distribution, which coincides with the effect of random perturbations. Compared to previous works, which suppress the Jacobian, we are the first to explicitly regularize for an *increased* Jacobian response.

3 Spectral properties of denoiser Jacobians

3.1 Generative denoising models

We use the term *generative denoising models* to refer to both diffusion (Ho et al., 2020) and flow-matching (Lipman et al., 2023). These models learn to transform Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ into data $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$ using a trained denoiser network that recovers information from noisy samples $\boldsymbol{x}_t$. In this work, we consider denoisers operating in the linear noise schedule (Ma et al., 2024)

$$\boldsymbol{x}_t = t\boldsymbol{x}_0 + (1-t)\boldsymbol{\epsilon}, \quad t \in [0, 1]. \tag{1}$$

To synthesize new data, we train the denoiser $\boldsymbol{f}_\theta$ using the objective

$$\mathcal{L}(\theta) = w(t) \left\| \boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \tag{2}$$

for randomly sampled batches $(\boldsymbol{x}_0, \boldsymbol{x}_t, t)$ of clean and noisy samples. We draw new samples, starting from $\boldsymbol{x}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and numerically solving the ODE

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{u}_\theta(\boldsymbol{x}_t, t), \quad \boldsymbol{u}_\theta(\boldsymbol{x}_t, t) = \frac{\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_t}{1-t} \tag{3}$$

with the solver operating backwards in time to produce $\boldsymbol{x}_0$ from the implicit distribution $p_\theta(\boldsymbol{x}_0)$ learned by the trained denoiser $\boldsymbol{f}_\theta$. Some design choices vary across different denoising models, such as noise schedule (Karras et al., 2022; Chen, 2023; Esser et al., 2024), velocity-prediction $\boldsymbol{u}$ instead of $\boldsymbol{x}_0$ (Li & He, 2026), and the per-timestep weighting $w(t)$ (Karras et al., 2022). Nevertheless, we can always obtain the denoising function $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$, the object of interest in this work.

3.2 Denoiser Jacobian properties

The denoiser Jacobian matrix $\boldsymbol{J}_\theta(\boldsymbol{x}_t, t) = \partial \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)/\partial \boldsymbol{x}_t$ maps local changes in the noisy input $\boldsymbol{x}_t$ to changes in the "clean" output $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$. These capture the learned *covariances* at each noise level; for instance, when changing a single pixel in the noisy input image, the Jacobian highlights the affected output pixels, revealing correlations learned by the model for the given image. This property is formally expressed by the relationship between the Jacobian and posterior covariance

$$\boldsymbol{J} = \frac{t}{(1-t)^2} \operatorname{Cov}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t). \tag{4}$$

For a derivation, refer to Appendix A. Although the identity only applies to the ideal denoiser case (Efron, 2011), it prompts us to analyze the Jacobian with tools used for covariance matrices, i.e. *spectral analysis*. By looking at the Jacobian, we examine the first-order properties of the denoiser (differences in denoising), which can provide more information than the zeroth-order denoising.

The spectral analysis of the Jacobian matrix requires computing the principal components, in other words, the eigenvectors corresponding to the directions of largest variance and the eigenvalues measuring expansion along each eigen-direction[1]. For image denoisers, the eigenvectors are images that show the maximal changes we can apply to the output image by perturbing the noisy input. The classical numerical method to compute eigenvectors and eigenvalues is power iteration, where we iteratively multiply a probe vector with the target matrix and re-normalize as $\boldsymbol{v}_{i+1} \leftarrow \boldsymbol{J}_\theta \boldsymbol{v}_i / \|\boldsymbol{J}_\theta \boldsymbol{v}_i\|$. Instead of a single probe that only recovers the largest eigenvector, we can use a simultaneous (or block) iteration (Clint & Jenning, 1970) to extract the top-n directions.

In the case of the deep neural network denoisers we consider, computing Jacobian-vector multiplications $\boldsymbol{J}_\theta \boldsymbol{v}_i$ is non-trivial as dimensions increase. To speed up the Jacobian-vector product computations, we use the finite-difference approximation along a direction $\boldsymbol{v}$

$$\boldsymbol{J}_\theta \boldsymbol{v} \approx \frac{\boldsymbol{f}_\theta(\boldsymbol{x}_t + \delta \boldsymbol{v}, t) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)}{\delta}. \tag{5}$$

[1] Although the Jacobian of a learned denoiser is not perfectly symmetric or positive semi-definite, we refer to the principal components as eigenvectors and consider eigenvalues to be real and positive.

Algorithm 1 Simultaneous iteration to compute top-n eigenvectors and eigenvalues of $\boldsymbol{J}_\theta$.

Input: Noisy sample $\boldsymbol{x}_t$, denoiser $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$, # of eigenvectors n, # of iterations K.

1: Starting basis $\boldsymbol{V}^{(0)} = [\boldsymbol{v}_1^{(0)} \ \dots \ \boldsymbol{v}_n^{(0)}]$ in $\mathbb{R}^n$, $\boldsymbol{v}_i^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
2: Obtain the factors $\boldsymbol{Q}^{(0)}\boldsymbol{R}^{(0)} = \boldsymbol{V}^{(0)}$ ▷ QR decomposition
3: **for** $k = 1, 2, \dots, K$ **do**
4: $\quad \boldsymbol{W} \leftarrow \boldsymbol{J}\boldsymbol{Q}^{(k-1)} = [\boldsymbol{J}\boldsymbol{q}_1^{(k-1)}, \dots, \boldsymbol{J}\boldsymbol{q}_n^{(k-1)}]$ ▷ $\boldsymbol{J}\boldsymbol{q}_i \approx (\boldsymbol{f}_\theta(\boldsymbol{x}_t + \delta\boldsymbol{q}_i, t) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t))/\delta$
5: $\quad$ Obtain the factors $\boldsymbol{Q}^{(k)}\boldsymbol{R}^{(k)} = \boldsymbol{W}$
6: **end for**

Return eigenvectors $\boldsymbol{V}^{(K)} = \boldsymbol{Q}^{(K)} = [\boldsymbol{v}_1^{(K)}, \dots, \boldsymbol{v}_n^{(K)}]$, eigenvalues $\|\boldsymbol{J}\boldsymbol{v}_1^{(K)}\|_2, \dots, \|\boldsymbol{J}\boldsymbol{v}_n^{(K)}\|_2$

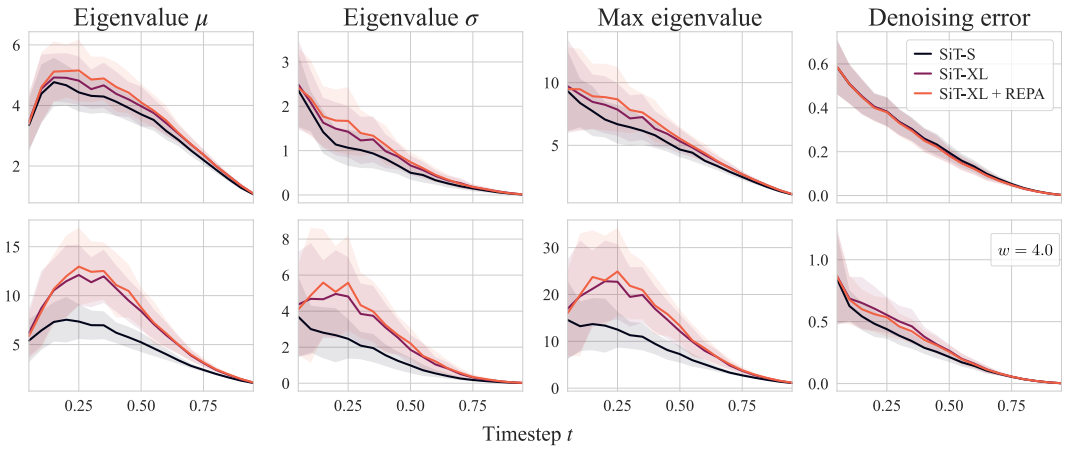


Figure 1: Using Algorithm 1 ($n = 10$, $K = 10$), we compute eigenvalues of Jacobians for different SiT denoisers. **Top**: We observe a distinct ordering that aligns with generative performance (SiT-S $<$ SiT-XL $<$ SiT-XL + REPA), whereas denoising capabilities are indistinguishable. **Bottom**: When using classifer-free guidance with $w = 4.0$ the differences between the models are accentuated.

This approximation is shown to be effective and provides a fast and accurate alternative to auto-differentiation (Graikos et al., 2026). Putting together the simultaneous power iteration and finite-difference approximation, we propose Algorithm 1 as a tool to find the top-n eigenvectors and corresponding eigenvalues of any denoiser Jacobian matrix $\boldsymbol{J}_\theta$. Importantly, this algorithm only requires forward passes through the model, which can be parallelized across eigenvectors.

3.3 Spectral Analysis of Pre-trained Denoiser Jacobians

We analyze the denoiser Jacobian of the state-of-the-art SiT-S, SiT-XL (Ma et al., 2024), and SiT-XL+REPA (Yu et al., 2025) generative denoising models. The results we obtain are comparable, as all three models were trained on ImageNet (Russakovsky et al., 2015) at 256×256 resolution and operate in the same VAE latent space (Rombach et al., 2022). We pick $t \in \{0.95, 0.90, \dots, 0.05\}$, and for each timestep, we randomly sample 100 images from the ImageNet validation set to construct noisy samples $\boldsymbol{x}_t$. We run all models on the same samples, with $\delta = 1$, $n = 10$ eigenvectors, and $K = 10$ iterations for Algorithm 1. We also re-run the analysis using classifier-free guidance (Ho & Salimans, 2022), which is frequently employed to improve sample quality, using guidance scale $w = 4$. In Figure 1, we plot descriptive statistics (mean, standard deviation, and maximum) of the measured eigenvalues, along with the mean squared error between denoised and real images.

We observe that the ordering in eigenvalue statistics ends up being the same as the ordering based on generative performance (FID (Heusel et al., 2017)); SiT-S $<$ SiT-XL $<$ SiT-XL+REPA. The Jacobian of better generative models has larger eigenvalues, notably over the initial and middle denoising steps. This clearly indicates that larger models not only denoise images more accurately but also use the additional parameters to capture more data variation, as reflected in the wider spectrum of their



Figure 2: Examples of a denoised image and the eigenvectors computed around it for pre-trained SiT models. We show the top-10 eigenvectors for the same image at $t = 0.4$. Better generative models capture more and sharper variability in their top components.

Jacobian. Our intuition is that while the smaller SiT-S synthesizes similar-looking images from all inputs in the neighborhood of $\boldsymbol{x}_t$, better-performing models produce vastly different results as small perturbations around $\boldsymbol{x}_t$ are amplified in the Jacobian. Interestingly, looking only at the denoising error is inconclusive, both with and without guidance.

We visualize eigenvectors across models in Figure 2. Showing $\boldsymbol{v}_i$ directly in latent space is uninformative, and instead, we visualize the effect of perturbing the noisy input $\boldsymbol{x}_t$ along $\boldsymbol{v}_i$, plotting the new prediction $\boldsymbol{f}_\theta(\boldsymbol{x}_t + \delta \boldsymbol{v}_i)$. The new predictions exhibit less variation for the SiT-S eigenvectors, compared to the larger models, which exhibit more meaningful changes. Again, looking only at the denoised image (left) gives no insights about the model's capability as an image generator.

4 Jacobian Regularization in Training

As shown above, the spectral properties of the denoiser Jacobian constitute strong evidence of the model's generation quality. A question that naturally arises is whether changing the Jacobian spectrum controls the model's generative capabilities. We revisit the denoising objective of Eq. (2), dropping the weight for simplicity, and add a second term that denoises a slightly perturbed input

$$\mathcal{L}(\theta) = \| \underbrace{\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0}_{\boldsymbol{r}} \|_2^2 + \tau \, \| \boldsymbol{f}_\theta(\boldsymbol{x}_t + \delta \boldsymbol{v}, t) - \boldsymbol{x}_0 \|_2^2 \tag{6}$$

where $\boldsymbol{v}$ is a perturbation direction. If $\boldsymbol{r}$ is the residual between the real and denoised image that the denoiser minimizes, the effect of perturbing the input is illustrated with the first-order approximation

$$\mathcal{L}(\theta) \approx \|\boldsymbol{r}\|_2^2 + \tau \, \|\boldsymbol{r} + \delta \boldsymbol{J}_\theta \boldsymbol{v}\|_2^2 \, . \tag{7}$$

The perturbed denoising term introduces the Jacobian explicitly in the training objective. Thus, we can regularize the Jacobian in a controlled manner. Different choices of the perturbation direction $\boldsymbol{v}$ regularize the denoiser Jacobian. We discuss these choices below, using the toy 2D mixture-of-Gaussians setting of Figure 3 (a). We provide implementation details in Appendix E.

Stochastic $\boldsymbol{v}$: For a random $\boldsymbol{v}$, with $\mathbb{E}[\boldsymbol{v}] = 0$, the expectation of Eq. (7) w.r.t. $\boldsymbol{v}$ gives us

$$\mathbb{E}_{\boldsymbol{v}}[\mathcal{L}(\theta)] = (1 + \tau) \, \|\boldsymbol{r}\|_2^2 + \tau \delta^2 \mathbb{E}_{\boldsymbol{v}} \left[\|\boldsymbol{J}_\theta \boldsymbol{v}\|_2^2 \right] . \tag{8}$$

The objective minimizes both the denoising loss and a Jacobian regularization term that penalizes expansion along the directions of $\boldsymbol{v}$. For a Normal $\boldsymbol{v}$, this regularization constrains the Jacobian expansion isotropically, reducing to a Frobenius norm regularizer $\mathbb{E}_{\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}[\|\boldsymbol{J}_\theta \boldsymbol{v}\|_2^2] = \|\boldsymbol{J}_\theta\|_F^2$ (Scarvelis & Solomon, 2024). To illustrate more fine-grained control over the directions in which we allow the Jacobian to expand, we choose $\boldsymbol{v} \in \{\pm[0, 1]^T, \pm[1, 0]^T\}^T$, which minimizes sample variation just along the horizontal and vertical axes. Figure 3 (b) shows how a denoiser trained with this regularization uses 'squares' to fit each Gaussian component.

Residual $\boldsymbol{v}$: To directly regularize eigenvalues, we would first have to compute them using a power iteration, as in Algorithm 1. Doing this at every training step is prohibitively slow; we want to design

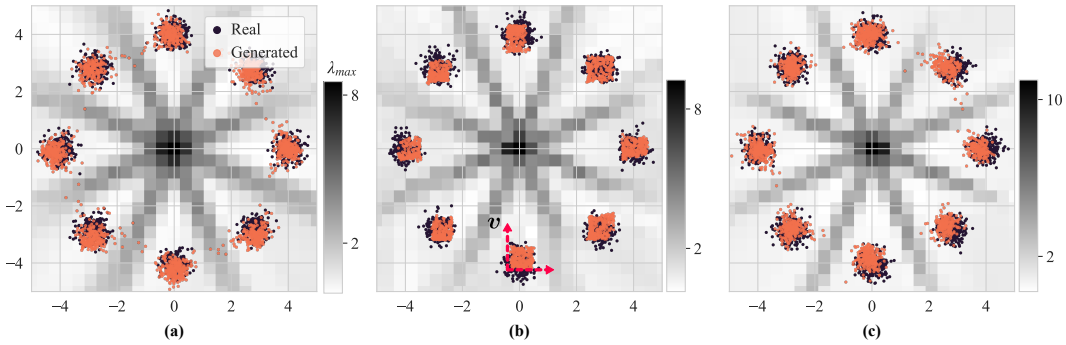


Figure 3: Training a denoising generative model on a mixture of 2D Gaussians. The grayscale color represents the maximum eigenvalue of the Jacobian at $t = 0.5$ for each point on the grid. **(a)** Baseline model trained without regularization. **(b)** Jacobian regularization using a perturbation that minimizes variation along the orthogonal axes. **(c)** Jacobian regularization using the residual perturbation, which increases eigenvalues (showing $\lambda_{\max}$), and results in fewer samples falling between modes.

a regularizer that increases the Jacobian response along its eigenvectors without having to compute them. Looking at Eq. (7), choosing $\boldsymbol{v} = -\boldsymbol{r} = \boldsymbol{x}_0 - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$ the objective becomes

$$\mathcal{L}(\theta) = \|\boldsymbol{r}\|_2^2 + \tau\|\boldsymbol{r} - \delta \boldsymbol{J}_\theta \boldsymbol{r}\|_2^2 \tag{9}$$

which minimizes both the denoising and the Jacobian regularization $\|\boldsymbol{r} - \delta \boldsymbol{J}_\theta \boldsymbol{r}\|$. This regularizer pushes the model to produce a Jacobian response along the direction of the residual such that $\boldsymbol{J}_\theta \boldsymbol{r} \approx \frac{1}{\delta}\boldsymbol{r}$. The prediction in $\boldsymbol{r}$ is used as a fixed target; we do not backpropagate through the perturbation.

With an appropriately small δ, we increase the expansion along the residual, which we expect to increase the overall Jacobian response. We empirically test this hypothesis in Figure 3 (c), where we train the denoiser using a fixed $1/\delta = 10$. The residual-regularized model exhibits larger eigenvalues in its Jacobian, resulting in fitting the target distribution better, with fewer samples between modes.

Why does the residual increase eigenvalues? The residual direction naturally emerges as an eigenvector-proxy regularization from Eq. (7) by imposing $\boldsymbol{J}_\theta \boldsymbol{r} \approx \frac{1}{\delta}\boldsymbol{r}$, and performs well in the 2D setting of Figure 3 (c). On the contrary, naively maximizing the Jacobian response in every direction does not guarantee an increase in the top eigenvalues (Appendix D.2) and can instead produce a model sensitive to all perturbations. To understand why the residual works well as a perturbation direction for regularization, we consider its covariance

$$\mathbb{E}\left[\boldsymbol{r}\boldsymbol{r}^T \mid \boldsymbol{x}_t\right] = \mathbb{E}\left[(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0)(\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0)^T \mid \boldsymbol{x}_t\right] = \mathrm{Cov}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \frac{(1-t)^2}{t}\boldsymbol{J} \tag{10}$$

which directly relates it to the Jacobian through the covariance of the denoiser posterior over $\boldsymbol{x}_0$ (Eq. 4). This allows us to quantify the alignment between the residual and Jacobian eigenvectors. We project the residual onto the eigenvector basis $\boldsymbol{v}_i$ of $\boldsymbol{J}$, for a given $\boldsymbol{x}_t$, as

$$\boldsymbol{r} = \sum_i \underbrace{\boldsymbol{v}_i^T \boldsymbol{r}}_{p_i} \boldsymbol{v}_i = \sum_i p_i \boldsymbol{v}_i \tag{11}$$

where p_i measures the overlap of the residual $\boldsymbol{r}$ with eigenvector $\boldsymbol{v}_i$. The residual covariance can be expressed using the eigenvector basis as

$$\mathbb{E}\left[\boldsymbol{r}\boldsymbol{r}^T \mid \boldsymbol{x}_t\right] = \mathbb{E}[\sum_{i,j} p_i p_j \boldsymbol{v}_i \boldsymbol{v}_j^T \mid \boldsymbol{x}_t] = \sum_{i,j} \mathbb{E}\left[p_i p_j \mid \boldsymbol{x}_t\right] \boldsymbol{v}_i \boldsymbol{v}_j^T. \tag{12}$$

Substituting in Eq. (10) and expanding the Jacobian into its eigenvalues λ_i and eigenvectors $\boldsymbol{v}_i$

$$\sum_{i,j} \mathbb{E}\left[p_i p_j \mid \boldsymbol{x}_t\right] \boldsymbol{v}_i \boldsymbol{v}_j^T = \frac{(1-t)^2}{t} \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T \tag{13}$$

where the RHS defines an orthonormal basis over $\mathbb{R}^N$, and we can derive that

$$\mathbb{E}\left[p_i p_j \mid \boldsymbol{x}_t\right] = \mathbf{1}[i=j]\frac{(1-t)^2}{t}\lambda_i \;\Rightarrow\; \mathbb{E}\left[p_i^2 \mid \boldsymbol{x}_t\right] = \frac{(1-t)^2}{t}\lambda_i. \tag{14}$$

The overlap p_i between the residual direction and an eigenvector is determined by the corresponding eigenvalue λ_i. By maximizing the Jacobian response along $\boldsymbol{r}$, we increase expansion more along the principal eigenvectors, which justifies why the residual regularizer produced a larger $\lambda_{\max}$ and did not result in an overly sensitive denoiser with bad generation quality (Appendix D.2).

5 IMAGENET EXPERIMENTS

The spectral analysis (Section 3.3) showed that in ImageNet-trained SiT models, the Jacobian spectrum is related to generative performance. Next, Section 4 proposed a regularizer that controls the Jacobian spectrum by decreasing or amplifying its response along random and eigen-directions, respectively. We next study whether applying this training regularization to ImageNet SiT models leads to the improvements in generation quality we observed in our initial analysis.

5.1 TRAINING SETUP

We use the SiT-S model and provide further results for SiT-B and UNet in Appendix B. We train the baseline with the $\boldsymbol{x}_0$-prediction objective of Eq. (2), per-timestep weighting $w(t) = 1/(1-t)^2$, and in the latent space of the SD-1.5 VAE (Rombach et al., 2022). We make two necessary modifications to facilitate ImageNet-scale training with the proposed Jacobian regularization.

First, although a fixed $1/\delta$ worked in the toy setting, our analysis of pre-trained models showed that eigenvalues varied significantly across timesteps (Figure 1). Therefore, we instead opt for an adaptive $1/\delta$, computed using the Rayleigh quotient (Horn & Johnson, 2012) along the residual direction $R(\boldsymbol{r}) = \boldsymbol{r}^T \boldsymbol{J}_\theta \boldsymbol{r} / \|\boldsymbol{r}\|^2$. We keep an exponential moving average of the Jacobian expansion across timesteps R_t, updated at every iteration using the finite-difference of Eq. (5) for $\boldsymbol{J}_\theta \boldsymbol{r}$. We set a multiplicative gain $1/\delta = kR_t$ during training – to simplify notation, we will refer to $1/\delta$ as the gain k, with larger $1/\delta$ imposing larger Jacobian eigenvalues.

Second, increasing the target gain $1/\delta$ amplifies regularization, but diminishes its gradient, as the perturbed $\boldsymbol{x}_t - \delta \boldsymbol{r}$ collapses to $\boldsymbol{x}_t$. To balance target gain and gradients, we mask the residual to retain only the top-5% pixels. By masking, the perturbed input explicitly contains less information, avoiding collapse. In Appendix C we discuss how masking does not change the regularization target, but instead strengthens its effect by a factor $\eta > 1$ determined by the masking ratio.

For the stochastic regularization, we perturb the input with randomly sampled Normal vectors with unit variance. Stochastic perturbations, compared to residual, have a stationary effect throughout training, making the main tuning parameter the weight τ instead of the radius $1/\delta$. We follow the same recipe as in the residual case, but now fix $1/\delta = 2$ and vary $\tau = \{1, 0.1\}$. We also train a model combining both perturbations by naively adding the regularization terms to the total loss. We use the best-performing hyperparameters of each and test whether the regularization effects are composable, and provide more details about the overall training process in Appendix F.

5.2 JACOBIAN REGULARIZATION CONTROLS SPECTRAL PROPERTIES

In Figure 4 we repeat the eigenvalue analysis on the baseline and models trained with residual regularization ($+\mathbf{v}_{\text{res}}$). We observe a clear increase in mean, standard deviation, and maximum eigenvalues over the baseline, which is again more pronounced in the earlier timesteps and when employing classifier-free guidance. Across the different target gains, a larger $1/\delta$ promotes overall larger eigenvalues, but we see diminishing returns when increasing gain from $5 \rightarrow 10$.

We also measure the eigenvalues of the stochastic regularization models ($+\mathbf{v}_{\text{rand}}$) in Figure 5. The regularization weight τ controls the strength of reduction in all eigenvalues as it minimizes $\|\boldsymbol{J}_\theta\|_F$. Classifier-free guidance shows that the $\tau = 1$ model is over-constrained, having smaller eigenvalues than the baseline model for its top components. In comparison, a moderate $\tau = 0.1$ gives the same spectral statistics as the baseline model, with the stochastic regularization reducing the Jacobian response along random, data-irrelevant directions and not affecting the principal components. The model trained with both regularization directions ($+\mathbf{v}_{\text{res}}$ $+\mathbf{v}_{\text{rand}}$ in Figure 5) exhibits similar eigenvalue statistics to the residual-only regularized model. Our interpretation is that the moderate stochastic regularization does not affect the regularization of the top eigen-directions that the residual imposes, but instead suppresses the spurious, data denoising-unrelated components.

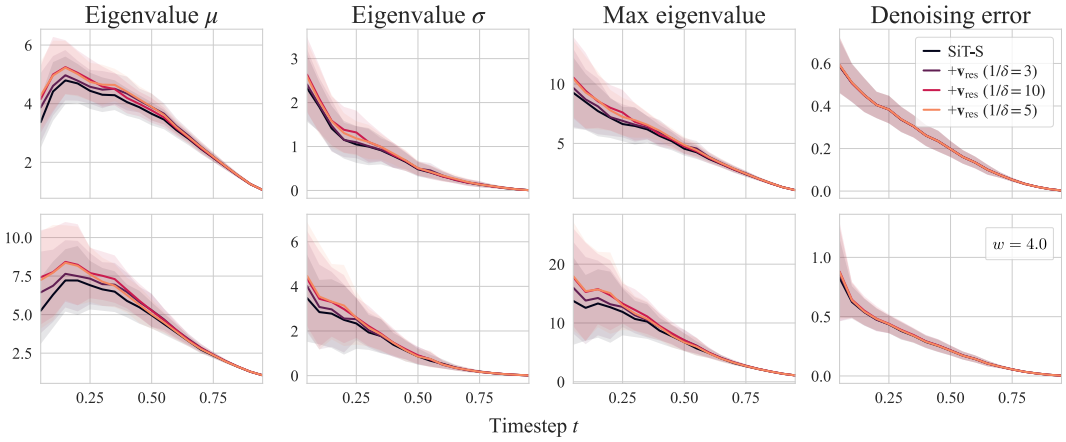


Figure 4: Eigenvalue analysis for SiT-S models trained with residual regularization. By varying the target gain $1/\delta$, we impose larger eigenvalues, with diminishing effects at $1/\delta = 10$. Using classifier-free guidance with $w = 4.0$ (bottom) amplifies the differences between the models.

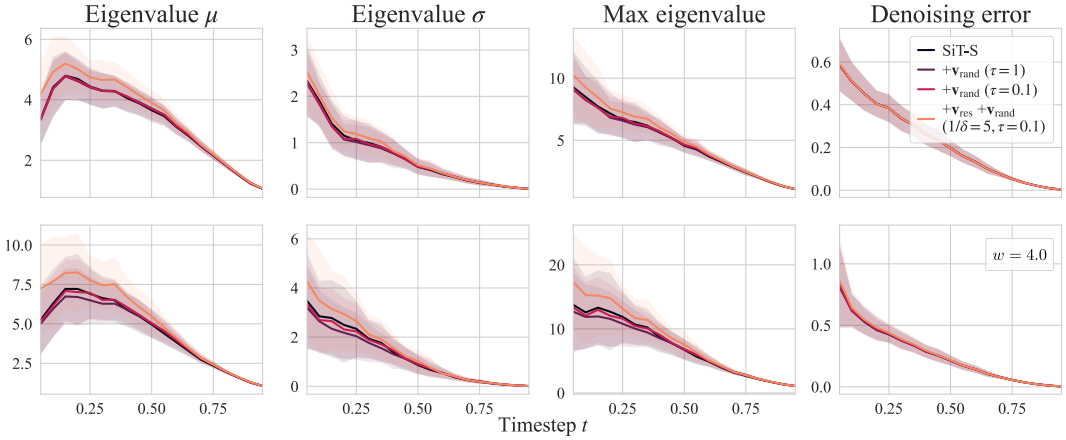


Figure 5: Eigenvalue analysis for SiT-S models trained with stochastic regularization. Using $\tau = 0.1$ does not alter the top-10 eigenvalues of the trained model. In contrast, with $\tau = 1.0$, the trained model is over-constrained, exhibiting smaller eigenvalues than the baseline model when using guidance with scale $w = 4.0$ (bottom). We also include the model combining both regularization signals ($1/\delta = 5$, $\tau = 0.1$), which we show obtains similar spectra to the residual-only regularized models.

5.3 Jacobian regularization improves generative performance

The eventual question is whether differences in spectrum are reflected in generative performance. In Figure 6 we calculate the FID during training for the SiT-S models, with and without regularization. First, we observe that $+\mathbf{v}_{\text{res}}$ improves FID, and the improvements align with the increase in spectra; the gain resulting in the largest eigenvalues ($1/\delta = 5$) also improves FID the most, and diminishing returns ($1/\delta = 10$) perform worse due to over-regularization. For the stochastic $+\mathbf{v}_{\text{rand}}$ regularization, $\tau = 1$ is worse than the baseline, penalizing useful eigenvectors and hurting generative performance. The more moderate $\tau = 0.1$, for which the top end of the spectrum did not change, yields an improved FID score between the baseline and the residual-regularized model.

Additionally, we test the baseline and best-performing regularized models in different settings in Table 1. The $+\mathbf{v}_{\text{res}}$ model benefits less from classifier-free guidance, showing smaller improvements than other models. We attribute this to increased sensitivity, amplified by classifier-free guidance, that can lead to larger error accumulation during sampling. This difference is notably reduced for fewer steps (20), hinting at sampling steps as the culprit. Finally, we observe small variations in fidelity (FID, precision) and diversity (recall) metrics among the models.

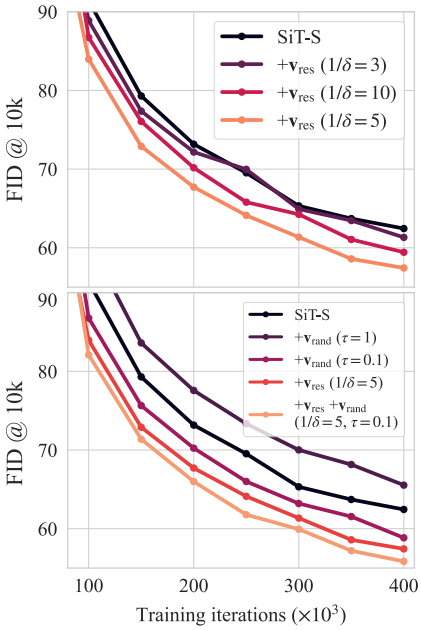


Figure 6: FID for the different regularizers and hyperparameters.

Table 1: Evaluation metrics varying sampling steps and guidance scales. Each regularization uses the best-performing model ($+\mathbf{v}_{\text{res}}$: $1/\delta = 5$), ($+\mathbf{v}_{\text{rand}}$: $\tau = 0.1$) and 50k images drawn with the Euler-Maruyama sampler.

Model	FID ↓	sFID ↓	Precision ↑	Recall ↑
250 steps, $w = 1.0$				
SiT-S	59.81	9.49	0.5929	0.3980
$+\mathbf{v}_{\text{res}}$	54.43	7.88	0.6053	0.3870
$+\mathbf{v}_{\text{rand}}$	56.12	8.86	0.6027	0.4177
$+\mathbf{v}_{\text{res}}$ $+\mathbf{v}_{\text{rand}}$	52.43	7.79	0.6202	0.4017
250 steps, $w = 1.5$				
SiT-S	30.56	6.91	0.5339	0.5642
$+\mathbf{v}_{\text{res}}$	29.19	7.86	0.5608	0.5489
$+\mathbf{v}_{\text{rand}}$	27.73	6.69	0.5411	0.5884
$+\mathbf{v}_{\text{res}}$ $+\mathbf{v}_{\text{rand}}$	27.10	7.74	0.5530	0.5678
20 steps, $w = 4.0$				
SiT-S	13.04	8.58	0.1887	0.8465
$+\mathbf{v}_{\text{res}}$	12.67	8.41	0.2062	0.8367
$+\mathbf{v}_{\text{rand}}$	12.78	8.19	0.1748	0.8690
$+\mathbf{v}_{\text{res}}$ $+\mathbf{v}_{\text{rand}}$	12.34	8.42	0.2051	0.8524

We close this discussion considering the model that combines both regularization objectives. The model achieves the best overall FID, suggesting that the regularizers act on complementary properties of the Jacobian spectrum. We interpret this as follows: the denoising objective identifies Jacobian directions from the data that are useful in reconstructing clean samples. The residual Jacobian regularizer further strengthens these task-relevant directions, while the stochastic regularizer attenuates weak directions that correspond to off-manifold or spurious variations. Our results suggest that the effect of regularization is spread across the entire Jacobian spectrum, with suppression of weak spurious components and amplification of useful, data-aligned directions.

6 Limitations

The residual is constructed from the initial model prediction, and thus requires a second forward pass through the denoiser. This considerably slows training compared to other regularization methods, which can usually be applied in a single forward pass. Furthermore, the increased sensitivity induced by the residual regularization weakens the benefits of classifier-free guidance. Although resolved when combining with the random perturbation, a better approach could make the regularizer timestep-dependent and only increase eigenvalues at timesteps where they are most beneficial.

7 Conclusion

In this work, we showed that the Jacobian of the denoisers used in diffusion and flow-matching models encodes useful information about their generative capabilities. Using spectral analysis, we showed how models that performed better in synthesizing samples from the learned distributions also have larger principal Jacobian eigenvalues. This observation motivated a regularization objective to control the Jacobian spectrum during training. This allowed us to test whether the relationship is bidirectional, whereby changing these spectral properties resulted in improved generation quality. Our findings have two main implications. First, we establish that the denoiser Jacobian provides a useful tool for identifying differences between models, with the proposed spectral analysis being just one of the ways to probe it. Second, identifying such differences informs the design of the training process itself, allowing us to impose the desirable denoiser properties in training, improving efficiency and performance of the model itself.

AI USE STATEMENT

In this work, we used generative AI tools (i) in creating and modifying the figures presented in the paper (ii) in helping to run experiments (iii) to identify and retrieve relevant literature and (iv) to aid or polish writing. We have not used generative AI tools to draft sections of the paper, generate synthetic datasets, or prove mathematical claims. We have reviewed all AI-assisted work by (i) verifying the LLM-generated code used in figures and visualizations (ii) verifying the LLM-generated code used in dataset pre-processing and loading (iii) extensively checking any suggested relevant works before deciding to cite them in our paper (iv) editing any LLM suggestions for refining and polishing text before using them in our text. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The methods used in the paper are described in Algorithms 1 and 2. Additionally, we provide code in the supplementary material that implements Algorithm 1 for the pre-trained SiT models. We also include code to replicate the mixture-of-Gaussians experiment of Section 4, and a mock implementation of Algorithm 2 on the ImageNet experiments in Section 5. Fully reproducing the ImageNet experiments requires downloading and pre-processing the entire dataset.

REFERENCES

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, 26, 2013.

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. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=HoThWhfxiK`.

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Maurice Clint and A Jenning. The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration. *The Computer Journal*, 13(1):76–80, 1970.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

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

Zach Evans, Cj Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12652–12665. PMLR, 21–27 Jul 2024.

Alexandros Graikos, Nebojsa Jojic, and Dimitris Samaras. Fast constrained sampling in pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 38:48205–48240, 2026.

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. *Advances in neural information processing systems*, 30, 2017.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Zahra Kadkhodaie, Florentin Guth, Eero Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *International Conference on Learning Representations*, volume 2024, pp. 46543–46567, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

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

Artem Lukoianov, Chenyang Yuan, Justin Solomon, and Vincent Sitzmann. Locality in image diffusion models emerges from data statistics. *Advances in Neural Information Processing Systems*, 38:95121–95157, 2026.

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*, pp. 23–40. Springer, 2024.

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *International Conference on Learning Representations*, volume 2024, pp. 49233–49263, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PmLR, 2021.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *arXiv preprint arXiv:2301.11706*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Christopher Scarvelis and Justin Solomon. Nuclear norm regularization for deep learning. *Advances in Neural Information Processing Systems*, 37:116223–116253, 2024.

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*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1185–1194. IEEE, 2025.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization. *arXiv preprint arXiv:2506.09027*, 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

A JACOBIAN AND COVARIANCE

The denoiser Jacobian is related to the covariance matrix of the learned denoising distribution: Let $\boldsymbol{x}_0$ denote the clean signal and let $\boldsymbol{x}_t$ be its noisy observation under additive Gaussian noise:

$$\boldsymbol{x}_t = t\boldsymbol{x}_0 + (1-t)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{15}$$

By Bayes' rule,

$$p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_t)}, \text{ where } p(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - t\boldsymbol{x}_0\|_2^2}{2(1-t)^2}\right). \tag{16}$$

Taking the logarithm of the posterior

$$\log p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \log p(\boldsymbol{x}_t \mid \boldsymbol{x}_0) + \log p(\boldsymbol{x}_0) - \log p(\boldsymbol{x}_t) \tag{17}$$

which, differentiating w.r.t. $\boldsymbol{x}_t$, gives

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t \mid \boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) = \frac{t\boldsymbol{x}_0 - \boldsymbol{x}_t}{(1-t)^2} - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t). \tag{18}$$

Tweedie's formula (Efron, 2011) states that

$$\underbrace{\mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right]}_{\hat{\boldsymbol{x}}_0} = \frac{1}{t}\boldsymbol{x}_t + \frac{(1-t)^2}{t}\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t). \tag{19}$$

or equivalently

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) = \frac{t\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_t}{(1-t)^2}. \tag{20}$$

Substituting this expression into Eq. (18)

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \frac{t\boldsymbol{x}_0 - \boldsymbol{x}_t}{(1-t)^2} - \frac{t\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_t}{(1-t)^2} = \frac{t(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)}{(1-t)^2} \tag{21}$$

which we can write as

$$\frac{\nabla_{\boldsymbol{x}_t} p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)}{p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)} = \frac{t(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)}{(1-t)^2}, \tag{22}$$

and hence

$$\nabla_{\boldsymbol{x}_t} p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) = \frac{t(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)}{(1-t)^2} p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t). \tag{23}$$

Now we can consider the Jacobian of the posterior mean:

$$\begin{aligned}
\boldsymbol{J} &= \frac{\partial \hat{\boldsymbol{x}}_0}{\partial \boldsymbol{x}_t} = \frac{\partial}{\partial \boldsymbol{x}_t}\mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right] = \frac{\partial}{\partial \boldsymbol{x}_t}\int \boldsymbol{x}_0\, p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\, d\boldsymbol{x}_0 \\
&= \int \boldsymbol{x}_0 \left[\nabla_{\boldsymbol{x}_t} p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\right]^T d\boldsymbol{x}_0 = \frac{t}{(1-t)^2}\int \boldsymbol{x}_0 (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\, d\boldsymbol{x}_0 \\
&= \frac{t}{(1-t)^2}\left(\int \boldsymbol{x}_0 \boldsymbol{x}_0^T p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\, d\boldsymbol{x}_0 - \int \boldsymbol{x}_0 \hat{\boldsymbol{x}}_0^T p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)\, d\boldsymbol{x}_0\right) \\
&= \frac{t}{(1-t)^2}\left(\mathbb{E}\left[\boldsymbol{x}_0 \boldsymbol{x}_0^T \mid \boldsymbol{x}_t\right] - \hat{\boldsymbol{x}}_0 \hat{\boldsymbol{x}}_0^T\right) \\
&= \frac{t}{(1-t)^2}\left(\mathbb{E}\left[\boldsymbol{x}_0 \boldsymbol{x}_0^\top \mid \boldsymbol{x}_t\right] - \mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right]\mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right]^T\right) \\
&= \frac{t}{(1-t)^2}\operatorname{Cov}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t).
\end{aligned} \tag{24}$$

This connects the Jacobian of an MMSE-optimal denoiser to the posterior covariance learned by the model

$$\boxed{\boldsymbol{J}_\theta = \frac{t}{(1-t)^2}\operatorname{Cov}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)}. \tag{25}$$

We can also differentiate Tweedie's formula directly,

$$\boldsymbol{J} = \frac{1}{t}\boldsymbol{I} + \frac{(1-t)^2}{t}\nabla^2_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) \tag{26}$$

which only tells us that the Jacobian matrix is symmetric, as the sum of $\boldsymbol{I}$ and a Hessian matrix. Finally, covariance matrices are always positive semi-definite, and thus we expect all eigenvalues of the (ideal) Jacobian matrix to be real and $\lambda_i \geq 0$.

B ADDITIONAL RESULTS

B.1 SPECTRAL ANALYSIS

We extend the analysis of pre-trained models by increasing the number of eigenvectors to $n = 20$ and using $K = 20$ iterations. The results, shown in Figure 7, retain the same ordering but with lower overall values. This is due to the exponentially decaying nature of the eigenvalues, where most of the energy is contained in the first few components.

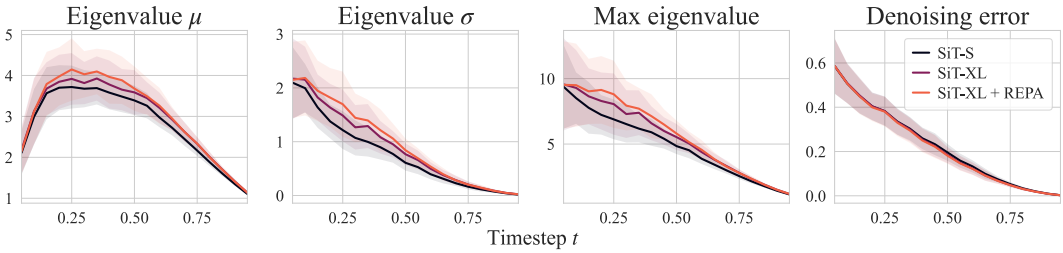


Figure 7: Using Algorithm 1, we measure eigenvalues of the Jacobians of different flow-matching denoisers. We find an ordering that correlates with the model's expected performance (SiT-S $<$ SiT-XL $<$ SiT-XL + REPA). **Top**: Using $n = 20$ and $K = 20$. **Bottom**: Using classifier-free guidance with scale $w = 4.0$.

We plot the spectrum of the top-100 eigenvalues for the trained SiT-S and SiT-S + Jacobian residual regularization models in Figure 8. We observe that the eigenvalues decay exponentially, which justifies limiting our analysis to the top-10 eigenvectors of each model in the main text. Furthermore, the increase in eigenvalues between the baseline and residual-regularized model is focused on the higher end of the spectrum, throughout all noise levels. This supports our analysis of the effect of the proposed residual direction, which we showed focuses on increasing the eigenvalues of directions with largest variance.

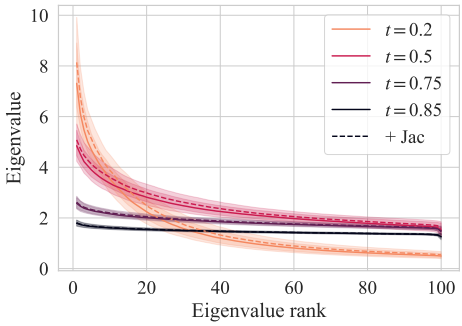


Figure 8: Comparing the spectrum of the top-100 eigenvalues of SiT-S (solid) and SiT-S with residual regularization (dashed).

B.2 EXTENSION TO OTHER MODELS

We test how the residual regularization scales with parameters by applying it to an SiT-B model (Ma et al., 2024). We also apply the residual Jacobian regularization to a UNet-based model (Ronneberger et al., 2015) to test whether the regularizer transfers to architectures beyond transformers. We utilize the UNet denoiser configuration from ADM (Nichol & Dhariwal, 2021) and set up a 30M-parameter model, comparable to the 33M parameters of the SiT-S model used in the main experiments. We again utilize the same VAE latent space as with SiT-S, training baseline and residual-regularized models with different gain targets.

When we run the eigenvalue analysis, we find that the same observations hold under both a larger model and a different denoiser architecture (Figure 9). The bigger SiT-B model saturates faster; when we increase the gain $1/\delta$ from 3→5, we observe the same diminishing returns we observed for SiT-S when increasing from 5→10. The UNet behaves similarly to the SiT-S, exhibiting comparable eigenvalue statistics.

We also evaluate the FID of the trained SiT-B and UNet denoisers in Figure 10. We observe similar improvements over the baseline as in the SiT-S model, with the FID score improving by ∼5 points between the baseline and the best-performing regularized model. As with the higher-gain case for the SiT-S model, the SiT-B model with $1/\delta = 5$ performs worse than the less-rigid $1/\delta = 3$, which suggests that the optimal $1/\delta$ choice depends on the model and not on the dataset. We provide examples of images generated with the SiT-B and SiT-B + Jacobian regularization models in Figure 21.

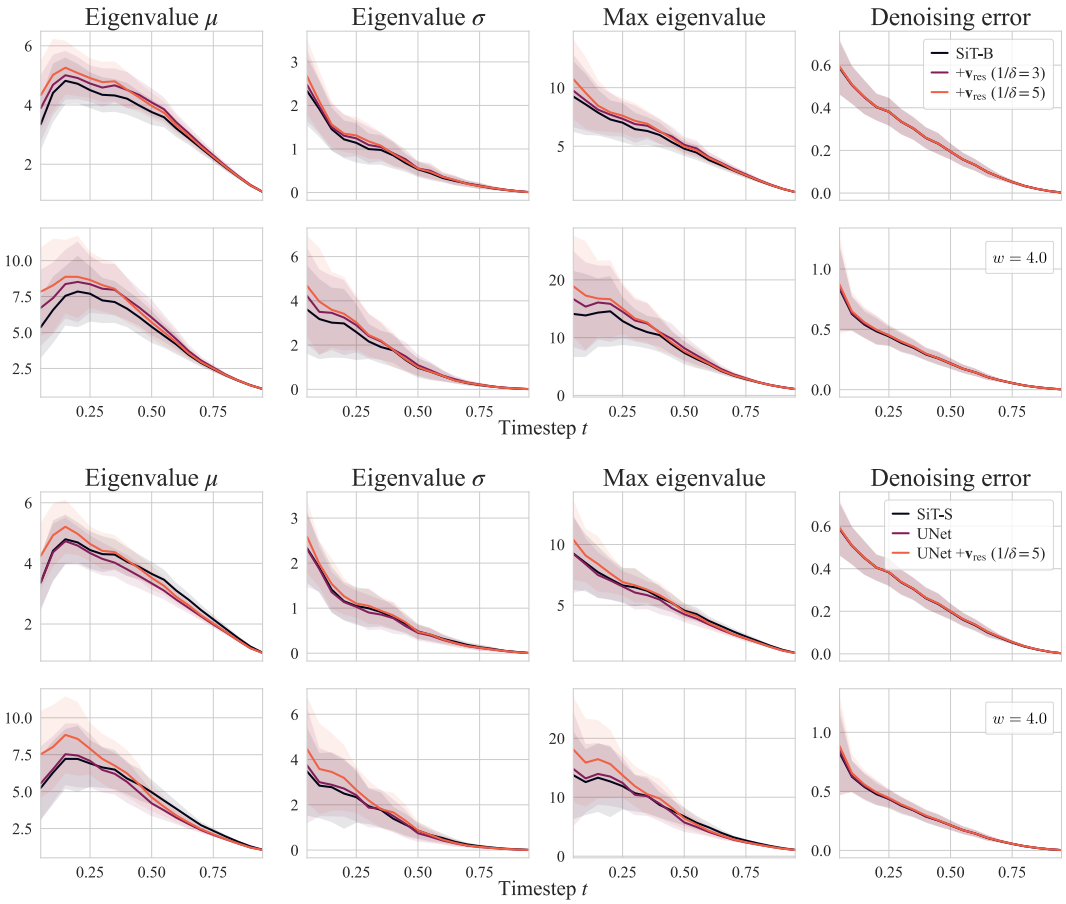


Figure 9: SiT-B and UNet denoiser eigenvalue comparison between baseline and models trained with the proposed residual regularization.

We gather all FID results in Table 2 to provide a complete picture of the evaluation. All models were evaluated by generating 50,000 images with the Euler-Maruyama sampler, using 250 inference steps and no classifier-free guidance. We compare the statistics of the generated images to the ImageNet statistics published by ADM (Nichol & Dhariwal, 2021). The findings are consistent across different models, with the residual regularization improving fidelity. Additionally, for the SiT-S model, we include the stochastic regularization and combined model results. We did not train a combined regularization variant for the SiT-B and UNet, but expect the performance improvements to be comparable.

C Residual and Masking

C.1 Effect of δ

In the main text, we used a masking operator on the residual to increase the 'push' on the eigenvalue increase without increasing the weight τ of the regularization term or reducing the step size δ. Regarding τ, we initially experimented with higher values but found that there was no significant effect. To understand the effect of masking, we will first discuss the effect of δ.

Using the regularization term with $\boldsymbol{v} = -\boldsymbol{r}$ as the perturbation direction, the loss we minimize is

$$\mathcal{L}(\theta) = \|\boldsymbol{r}\|_2^2 + \tau \underbrace{\|\boldsymbol{r} - \delta \boldsymbol{J}\boldsymbol{r}\|_2^2}_{\mathcal{L}_{\text{reg}}}. \tag{27}$$

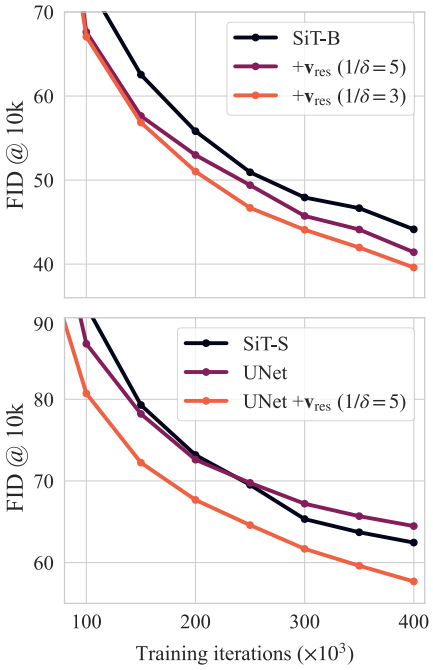


Figure 10: FID comparisons for an SiT-B and a UNet-based denoiser trained with and without Jacobian regularization. For the UNet, we include the baseline SiT-S FID as a reference curve.

Table 2: FID results for all trained models. We evaluate the EMA checkpoint of each model at 400k training iterations with 50,000 generated samples. We use the Euler-Maruyama sampler, 250 inference steps, and no classifier-free guidance. The two hyperparameters we control are the residual gain target ($1/\delta$) and the regularization weight for the random perturbations (τ).

Model	FID @ 50k ↓
SiT-S	59.78
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 2$)	59.86
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 3$)	58.88
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 5$)	54.43
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 10$)	56.75
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 5$, Full)	59.14
+$\mathbf{v}_{\text{rand}}$ ($\tau = 1$)	63.19
+$\mathbf{v}_{\text{rand}}$ ($\tau = 0.1$)	56.07
+$\mathbf{v}_{\text{rand}}$ +$\mathbf{v}_{\text{res}}$ ($\tau = 0.1$, $1/\delta = 5$)	52.43
SiT-B	41.12
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 3$)	36.82
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 5$)	38.68
UNet	60.76
+$\mathbf{v}_{\text{res}}$ ($1/\delta = 5$)	55.10

Focusing on the regularization, we can first project the residual onto the eigenvectors of the Jacobian, as in $\boldsymbol{r} = \sum_i p_i \boldsymbol{v}_i$. We then rewrite the regularizer as

$$\mathcal{L}_{\text{reg}}(\theta) = \left\| \sum_i p_i \boldsymbol{v}_i - \delta \boldsymbol{J} \sum_i p_i \boldsymbol{v}_i \right\|_2^2 = \left\| \sum_i p_i (1 - \delta\lambda_i) \boldsymbol{v}_i \right\|_2^2 = \sum_i p_i^2 (1 - \delta\lambda_i)^2 \tag{28}$$

Taking the gradient of the regularization term w.r.t. an eigenvalue λ_i gives us

$$\frac{\partial \mathcal{L}_{\text{reg}}}{\partial \lambda_i} = -2p_i^2 \delta (1 - \delta\lambda_i) \tag{29}$$

which has two competing factors. By decreasing δ, we push λ_i towards a higher value through $\lambda_i = 1/\delta$, but simultaneously we attenuate the effect of the regularization term since the gradient is multiplied by δ itself. Therefore, it is impractical to pursue higher eigenvalues in the model only by making delta smaller. By introducing masking, we find a workaround that achieves the desired effect without diminishing the effect of the regularization term.

To formulate this, we consider masking as the linear operator $\boldsymbol{M}$, which makes the loss of Eq 9

$$\mathcal{L}(\theta) = \|\boldsymbol{r}\|_2^2 + \tau \|\boldsymbol{r} - \delta \boldsymbol{J}\boldsymbol{M}\boldsymbol{r}\|_2^2. \tag{30}$$

What the regularization term does is push the residual towards

$$\boldsymbol{J}\boldsymbol{M}\boldsymbol{r} = \frac{1}{\delta}\boldsymbol{r}. \tag{31}$$

The minimum rank positive semi-definite $\boldsymbol{J}$ that satisfies this (Horn & Johnson, 2012) is

$$\boldsymbol{J}_{\min} = \frac{1}{\delta \boldsymbol{r}^T \boldsymbol{M}\boldsymbol{r}} \boldsymbol{r}\boldsymbol{r}^T = \frac{1}{\delta \|\boldsymbol{M}\boldsymbol{r}\|^2} \boldsymbol{r}\boldsymbol{r}^T \tag{32}$$

and applying this to the full residual would look like

$$\boldsymbol{J}_{\min}\boldsymbol{r} = \frac{1}{\delta} \frac{\|\boldsymbol{r}\|^2}{\|\boldsymbol{M}\boldsymbol{r}\|^2} \boldsymbol{r} = \frac{1}{\delta}\eta\boldsymbol{r}, \quad \text{where } \eta = \frac{\|r\|^2}{\|\boldsymbol{M}\boldsymbol{r}\|^2} > 1. \tag{33}$$

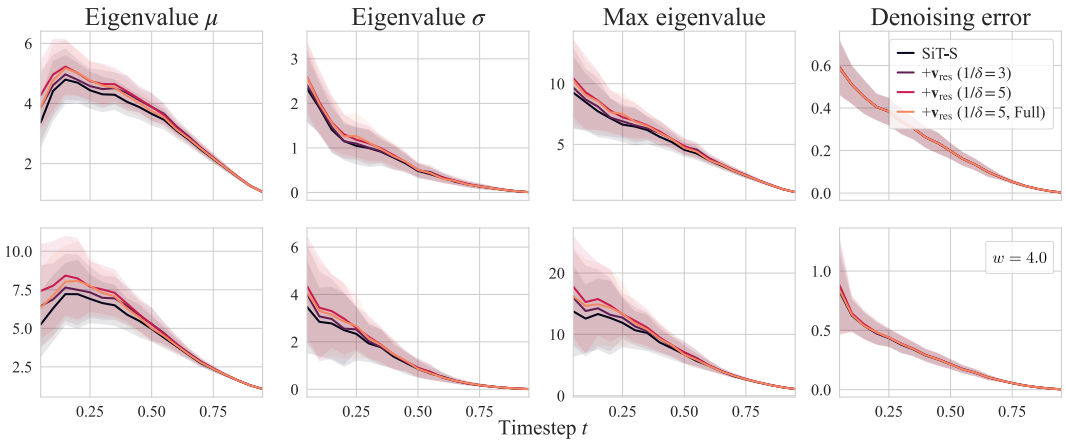


Figure 11: We compare trained model eigenvalues when using the masked and full residual as the perturbation direction. **Top**: Comparison for $n = 10$, $K = 10$. **Bottom**: Same comparison using classifier-free guidance with scale $w = 4.0$.

This shows that by masking the residual, we still apply the same regularization term, but now the target gain $1/\delta$ is also multiplied by a factor $\eta > 1$, where η depends on the masking ratio (more masking $\rightarrow$larger η). For our experiments, we choose a mask that retains the top-5% residual pixels by average intensity.

C.2 Training without masking

We train an SiT-S using the full residual and the best-performing $1/\delta = 5$. In Figure 11 we compare the eigenvalues from different models trained with masking and the non-masked model. The first observation we make is that for the same radius $1/\delta = 5$, the masked model Jacobian has larger eigenvalues, especially for lower timesteps, where the full residual has similar mean eigenvalues to a masked model using $1/\delta = 3$. This observation is consistent with the result of Eq. 33, which predicts that for the same $1/\delta$, the masked residual will produce higher eigenvalues.

When we compare the FID achieved by each of the models (Figure 12), we see that the SiT-S regularized with the full residual performs similarly to the masked residual model using $1/\delta = 3$. However, it would be naive to claim that the eigenvalue statistics alone determine generative performance; a model with a massive eigenvalue that has memorized images for each ImageNet class would not attain a competitive FID score. Thus, we hypothesize that qualitative differences also emerge in the larger and Jacobian-regularized models.

In Figure 13, we probe this hypothesis, showing different Jacobian-residual products. The best-performing model (SiT-XL + REPA) encodes plausible structures for the clean image in the direction of the residual, while the SiT-S and SiT-S + Full residual models produce blurrier results. The masked residual model differs from the other SiT-S variants, as it exhibits a result that is qualitatively closer to the SiT-XL structured Jacobian-residual product. This backs our overall suggestion that the spectral properties we discuss in this paper are but one of the directions from which we can approach the first-order properties of the denoiser.

C.3 Comparing residual and Stochastic regularization

The stochastic regularized model with $\tau = 0.1$ considerably improves over the baseline, while simultaneously not showing a different Jacobian spectrum, as measured in Figure 5. In Figure 14 we qualitatively compare generated images from the baseline SiT-S and the stochastic or residual-regularized models. The residual perturbation induces larger corrections on the baseline-generated images, which indicates that this regularization has a stronger effect than random. Despite improving FID, this has also potentially harmful effects, as over-correction can make images unrealistic. We believe that this is the effect that leads to smaller improvements with classifier-free guidance (Section 5.3 and Table 1).

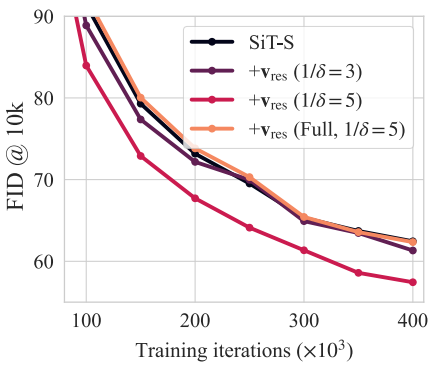


Figure 12: We compare the FID between models trained with the full and masked Jacobian regularization. The full residual regularization, with $1/\delta = 5$, behaves similarly to the masked residual using $1/\delta = 3$.

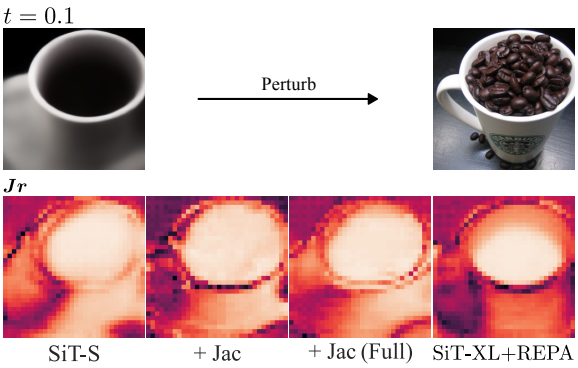


Figure 13: Qualitative comparison of the Jacobian-residual product. SiT-XL + REPA not only has larger eigenvalues (Figures 1 vs. 11) but also captures different spatial structures. While the base SiT-S and the SiT-S+Full are similar, the model trained with masking highlights noticeably different structures.



Figure 14: We synthesize images from the same noise, using 50 steps and classifier-free guidance scale $w = 4.0$ to amplify differences. We observe that the model using the residual perturbation performs larger corrections to the baseline-generated images, indicating a stronger overall effect.

D Direct Jacobian regularization

D.1 Training

Instead of using the perturbed regression objective of Eq (9) to regularize the Jacobian of the trained denoiser, we can directly use the finite-difference approximation

$$\boldsymbol{J}_\theta \boldsymbol{v} \approx \frac{f_\theta(\boldsymbol{x}_t + \delta \boldsymbol{v}, t) - f_\theta(\boldsymbol{x}_t, t)}{\delta} \tag{34}$$

whereby choosing the appropriate $\boldsymbol{v}$ and the sign (minimize/maximize), we control the regularization, e.g., $\boldsymbol{v}\ \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for suppressing diagonal directions or $\boldsymbol{v} = -\boldsymbol{r}$ to boost the eigenvalues. For the residual direction, we get the objective

$$\mathcal{L}(\theta) = \|f_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2 + \tau \|f_\theta(\boldsymbol{x}_t - \delta \boldsymbol{r}, t) - f_\theta(\boldsymbol{x}_t, t)\|_2^2. \tag{35}$$

We train an SiT-S model with this objective, using $1/\delta = 5$ and $\tau = 0.0005$; for any larger τ we tried, the loss diverged. The resulting denoiser achieves an FID of ∼65, considerably higher

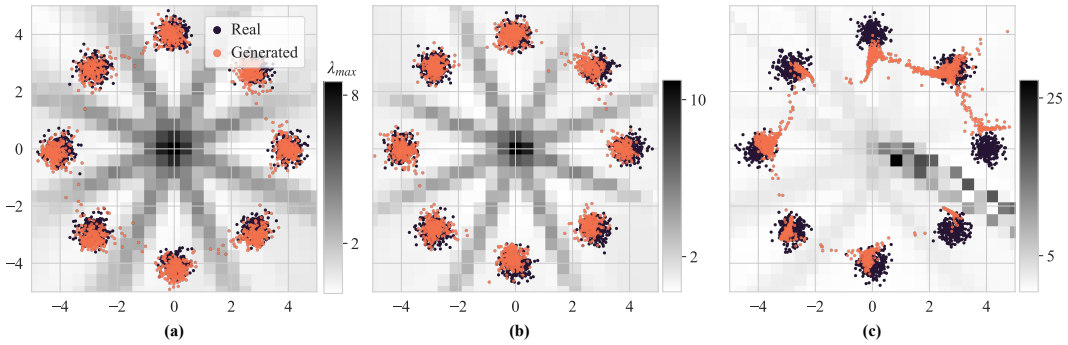


Figure 15: Training a denoising generative model on a mixture of 2D Gaussians. The grayscale color represents the maximum eigenvalue of the Jacobian at $t = 0.5$ for each point on a 2D grid. **(a)** Baseline model trained without regularization. **(b)** Jacobian regularization using the residual perturbation, pushing $\lambda_{\max}$ higher. **(c)** Direct regularization (Appendix D) maximizing all eigenvalues simultaneously. The resulting model is overly sensitive and fails to learn the target distribution.

than the 59 achieved by the baseline SiT. We hypothesize that this direct Jacobian regularization is more prone to noise in training and largely unstable. The regularization proposed in the main text is grounded in the base denoising regression objective, which, even when it doesn't provide any additional signal (e.g., when using a small $1/\delta$), does not hurt the overall training.

D.2 Residual vs. Normal expansion

We can contrast the result of the residual direction, which we showed overlaps with eigenvectors proportionally to the corresponding eigenvalue (Eq. 14), by analyzing the case where we maximize Jacobian expansion along random directions $\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Using the same approach, we can analyze the covariance of the perturbation direction

$$\mathbb{E}\left[\boldsymbol{e}\boldsymbol{e}^T \mid \boldsymbol{x}_t\right] = \mathbb{E}\left[(\boldsymbol{e}-\mathbf{0})(\boldsymbol{e}-\mathbf{0})^T\right] = \mathrm{Cov}(\boldsymbol{e}) = \boldsymbol{I}. \tag{36}$$

If we project a random vector $\boldsymbol{e}$ on the eigenvectors $\boldsymbol{v}_i$ of $\boldsymbol{J}$ we would get

$$\boldsymbol{e} = \sum_i \underbrace{\boldsymbol{v}_i^T \boldsymbol{e}}_{q_i} \boldsymbol{v}_i = \sum_i q_i \boldsymbol{v}_i \tag{37}$$

which allows us to write the covariance as

$$\mathbb{E}\left[\boldsymbol{e}\boldsymbol{e}^T \mid \boldsymbol{x}_t\right] = \sum_i \sum_j \mathbb{E}\left[q_i q_j \mid \boldsymbol{x}_t\right] \boldsymbol{v}_i \boldsymbol{v}_j^T. \tag{38}$$

In this case, however, we have no identity that connects the covariance to the Jacobian, giving instead

$$\sum_{i,j} \mathbb{E}\left[q_i q_j \mid \boldsymbol{x}_t\right] \boldsymbol{v}_i \boldsymbol{v}_j^T = \boldsymbol{I} = \sum_i 1 \cdot \boldsymbol{v}_i \boldsymbol{v}_i^T \tag{39}$$

which, by using the same RHS orthonormal basis argument, leads to

$$\mathbb{E}\left[q_i q_j \mid \boldsymbol{x}_t\right] = 1 \;\Rightarrow\; q_i^2 = 1. \tag{40}$$

Thus, using a randomly sampled Normal direction to maximize Jacobian expansion increases all eigenvalues simultaneously. In contrast, the residual direction focuses on the largest eigenvalues, which we hypothesize helps avoid introducing uncontrolled noise to the trained model, as it does not increase sensitivity in random directions. Furthermore, the proposed perturbed-input regularization does not naturally lead to this isotropic objective. Implementing this regularizer would require the direct Jacobian probing method, which we established before as subpar. We provide an example in Figure 15, showing how this isotropic maximization leads to a bad generative model.

Algorithm 2 Training iteration with Jacobian regularization

Input: Denoiser $\boldsymbol{f}_\theta(\boldsymbol{x}_t, t)$, sample $\boldsymbol{x}_0$, spectrum estimate R_t, EMA decay α, regularizer weight τ.

Denoising loss

$\boldsymbol{x}_t = t\boldsymbol{x}_0 + (1-t)\boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$w(t) = 1/(1-t)^2$

$\mathcal{L}_{\text{denoise}} = w(t)\,\|\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2$

Jacobian Regularization

$\boldsymbol{r} = \text{sg}[\boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0]$

if masking **then**

    $\boldsymbol{M} = \mathbf{1}[\boldsymbol{r} \geq \mathcal{Q}_{0.95}(\boldsymbol{r})]$

    $\boldsymbol{r} = \boldsymbol{M}\boldsymbol{r}$

end if

$\delta_t = 1/(kR_t)$

$\mathcal{L}_{\text{reg}} = w(t)\,\|\boldsymbol{f}_\theta(\boldsymbol{x}_t - \delta_t\boldsymbol{r}, t) - \boldsymbol{x}_0\|_2^2$

Update spectrum estimate

$\boldsymbol{J}_\theta\boldsymbol{r} = (\boldsymbol{f}_\theta(\boldsymbol{x}_t - \delta\boldsymbol{r}, t) - \boldsymbol{f}_\theta(\boldsymbol{x}_t, t))/(-\delta)$

$R_t = \alpha R_t + (1-\alpha)(\boldsymbol{r}^T\boldsymbol{J}_\theta\boldsymbol{r})/\|\boldsymbol{r}\|_2^2$

Return $\mathcal{L} = \mathcal{L}_{\text{denoise}} + \tau\mathcal{L}_{\text{reg}}$

E TOY EXPERIMENT

In the toy experiment, we train a simple MLP denoiser with three hidden 128-dim layers and SiLU activations. We encode the timestep with 64-dim sinusoidal embeddings, and use a zero-initialized output linear layer to map to the 2D $\boldsymbol{x}_0$ prediction. Each model is trained for 10,000 iterations, with a learning rate of 0.002 and a batch size of 512, using the x-prediction and $w(t) = 1/(1-t)^2$ (equivalent to the v-loss formulation of Li & He (2026)).

For the "square" regularization, we randomly sample $\boldsymbol{v} = \{\pm[0,1]^T, \pm[1,1]^T\}$ and set $\delta = 1$ and $\tau = 0.1$. For the residual regularization, we use $1/\delta = 10$ and $\tau = 1$. To visualize the largest eigenvalue, we use backwards differentiation to construct the 2×2 Jacobian matrix for every point and use SVD to get the largest singular value. As mentioned in our initial discussion, in all our experiments we consider the Jacobian to be symmetric, and thus use the singular value and eigenvalue terms interchangeably.

F IMAGENET TRAINING

In Algorithm 2 we describe a training iteration with the residual regularization. The regularization loss can only be computed *after* the initial denoising prediction since it relies on the model output to construct the residual perturbation direction. For our ImageNet experiments, we construct the mask $\boldsymbol{M}$ by keeping the top-5% of pixels in the residual. For stochastic regularization, we follow the same training scheme, exchanging the residual direction with a randomly sampled Normal vector.

We train all models using the same setup, setting the learning rate to 0.0002, clipping gradient norms to 1.0, and using a model EMA with decay 0.9999. For the Rayleigh quotient estimate, we set the decay $\alpha = 0.99$, roughly updating our estimates every 100 steps per t. We pre-extract VAE latents for all images, using the SD-1.5 VAE as with the base SiT models.

We choose the Rayleigh quotient for the eigenvalue target, instead of the raw gain $\|\boldsymbol{J}_{theta}\boldsymbol{r}\|/\|\boldsymbol{r}\|$, as the Rayleigh quotient measures expansion parallel to the direction and ignores orthogonal changes. To visualize this difference, in Figure 16 we plot the gain $\|\boldsymbol{J}\boldsymbol{r}\|/\|\boldsymbol{r}\|$ and Rayleigh quotient $\boldsymbol{r}^T\boldsymbol{J}\boldsymbol{r}/\|\boldsymbol{r}\|^2$ measured during the training of the Jacobian-regularized SiT-S/B. For numerical stability ($\|\boldsymbol{r}\| \to 0$ as $t \to 1$), we set both to 1 for $t > 0.85$. The gain overestimates expansion along the residual direction in early timesteps, where small changes in the direction of the residual have a

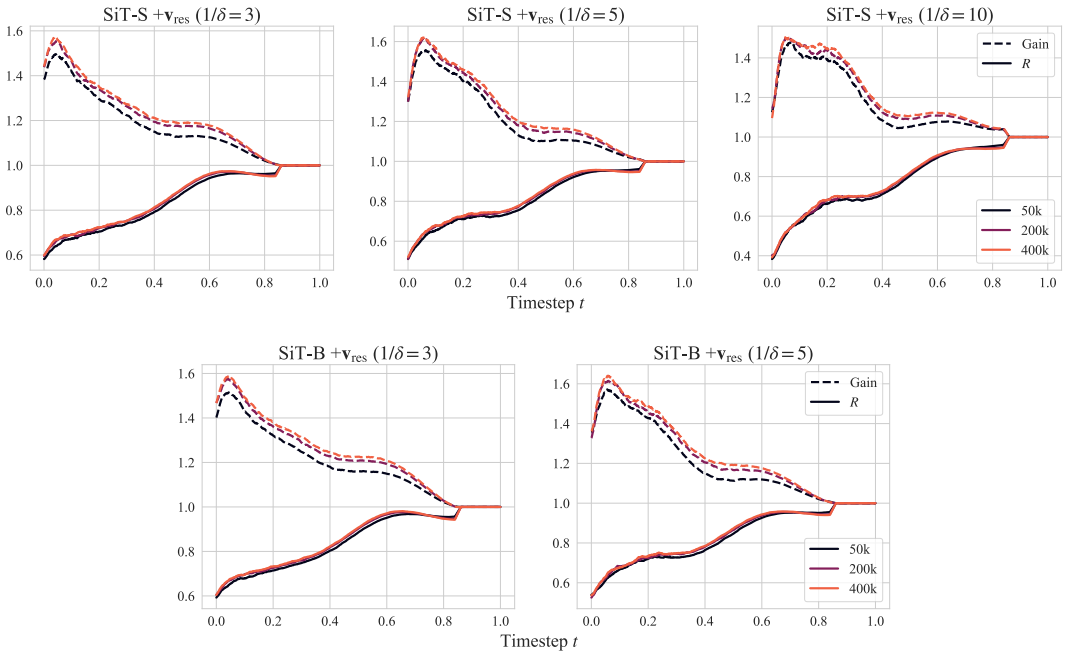


Figure 16: Gain and Rayleigh quotient R during training for Jacobian-regularized SiT-S/B models. The gain overestimates the effect of the residual perturbation gain. We use R to set the gain target during training.

large orthogonal component that is unrelated to the eigen-directions. Experimentally, we found the Rayleigh quotient to be the more stable training approach.

Instead of modifying the target gain $1/\delta$, we initially increased τ, and found the effect to be limited, as it only makes the regularization denoising have a larger gradient than the base denoising loss, which is dominated by the minimization of the residual. We set $\tau = 1$ for all our residual Jacobian regularization experiments. For stochastic perturbations, τ is the main control signal, as the gain/Rayleigh quotient (which, in expectation, are equivalent) is fixed throughout training.

A dynamic that is important to mention is how the perturbed input $\boldsymbol{x}_t - \delta \boldsymbol{r}$ is easier for the denoiser to regress, since the residual $\boldsymbol{r} = \boldsymbol{f}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0$ introduces the target image $\boldsymbol{x}_0$ into the input. Using a larger δ makes the regularization task too simple, as we include more and more of the target $\boldsymbol{x}_0$ in the input. On the other hand, by making δ too small, the task collapses to the base denoising loss. The online tuning we perform for the residual regularization is necessary to find the balance between simplifying the perturbed regression task and making it uninformative. In contrast to that, the stochastic regularization effect is stationary throughout training, and the response of the model to stochastic perturbations does not change as denoising improves.

We visualize eigenvectors during training of the SiT-B model in Figure 17. During training, we can see the progression of the eigenvectors getting sharper and adding more variations to the directions. Moreover, we compare the eigenvectors for the baseline SiT-B and residual-regularized model in Figure 18, where although the difference is less prominent when comparing SiT-XL to SiT-S, we see similar patterns emerge.

G Visualizing eigenvectors

The models we utilize all operate in the latent space of the SD-1.5 VAE (Rombach et al., 2022). To visualize the eigenvectors we compute, we must first decode them into image space, so that the changes that the eigenvector is pointing towards can be meaningfully shown as image pixels.

To decode latent eigenvectors to pixels, we first scale them by $s = 100$, as decoding unit-norm latents through the decoder does not produce meaningful images, and also remove the decoder's



Figure 17: Examples of a denoised image and the eigenvectors computed around it for different training iterations in the SiT-B + residual regularization model. We show the top-10 eigenvectors for the same image at $t = 0.4$. As training progresses, the model captures wider and sharper variations in its top components.



Figure 18: Examples of a denoised image and the eigenvectors computed around it for the SiT-B model with and without residual regularization. We show the top-10 eigenvectors for the same image at two different noises for $t = 0.4$. The residual-regularized model improves the spectrum by making larger changes with top components.

zero-bias since decoding a zero latent does not correspond to a zero image. We can express that as $\mathcal{D}(\boldsymbol{f}(100\boldsymbol{v}_i, t)) - \mathcal{D}(\mathbf{0})$, where $\mathcal{D}$ involves both the decoding and re-scaling to valid pixel values. In addition to eigenvectors, a more helpful visualization is to show the effect of perturbing the input towards the eigen-direction, i.e., $\boldsymbol{f}(\boldsymbol{x}_t + \delta\boldsymbol{v}_i, t)$.

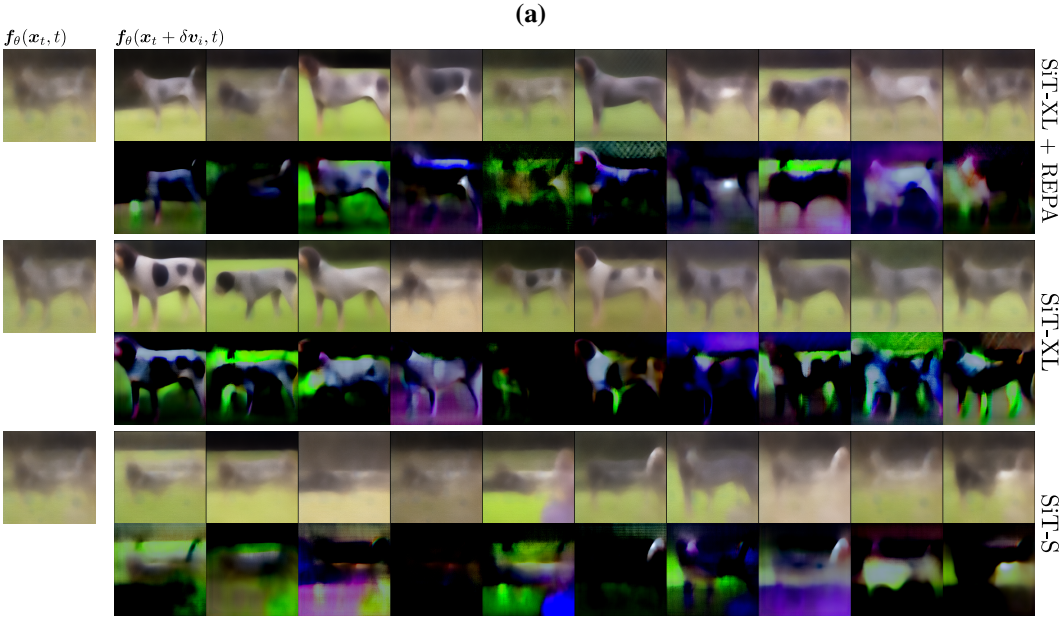
We provide examples of both the eigenvectors and the outputs of the denoiser in Figure 19. We first show eigenvectors at $t = 0.4$ and guidance $w = 4.0$, which produces larger eigenvectors for all models. We also visualize eigenvectors at $t = 0.2$, where now the variations capture larger-scale structures in the image.

H Denoiser features

In Figure 20, we visualize the denoiser features for the base and the Jacobian-regularized SiT-B models. We L_2-normalize each transformer block's features and project them into RGB using PCA, fitting the principal components to each block separately. We observe that global structures seem to emerge in earlier blocks in the Jacobian-regularized model. For instance, in the $t = 0.3$ row, the outline of the TV or the shape of the dog's head appears around block 3 for the Jacobian-regularized model, whereas it is not clearly visible until block 5 for the base SiT-B.



(a)



(b)

Figure 19: **(a)** Eigenvectors at $t = 0.4$, using classifier-free guidance with scale $w = 4.0$. Increasing guidance leads to amplified differences between the eigenvectors and larger eigenvalues. **(b)** Eigenvectors at $t = 0.2$ without classifier-free guidance. The variations in lower timesteps capture larger-scale structures in the images.

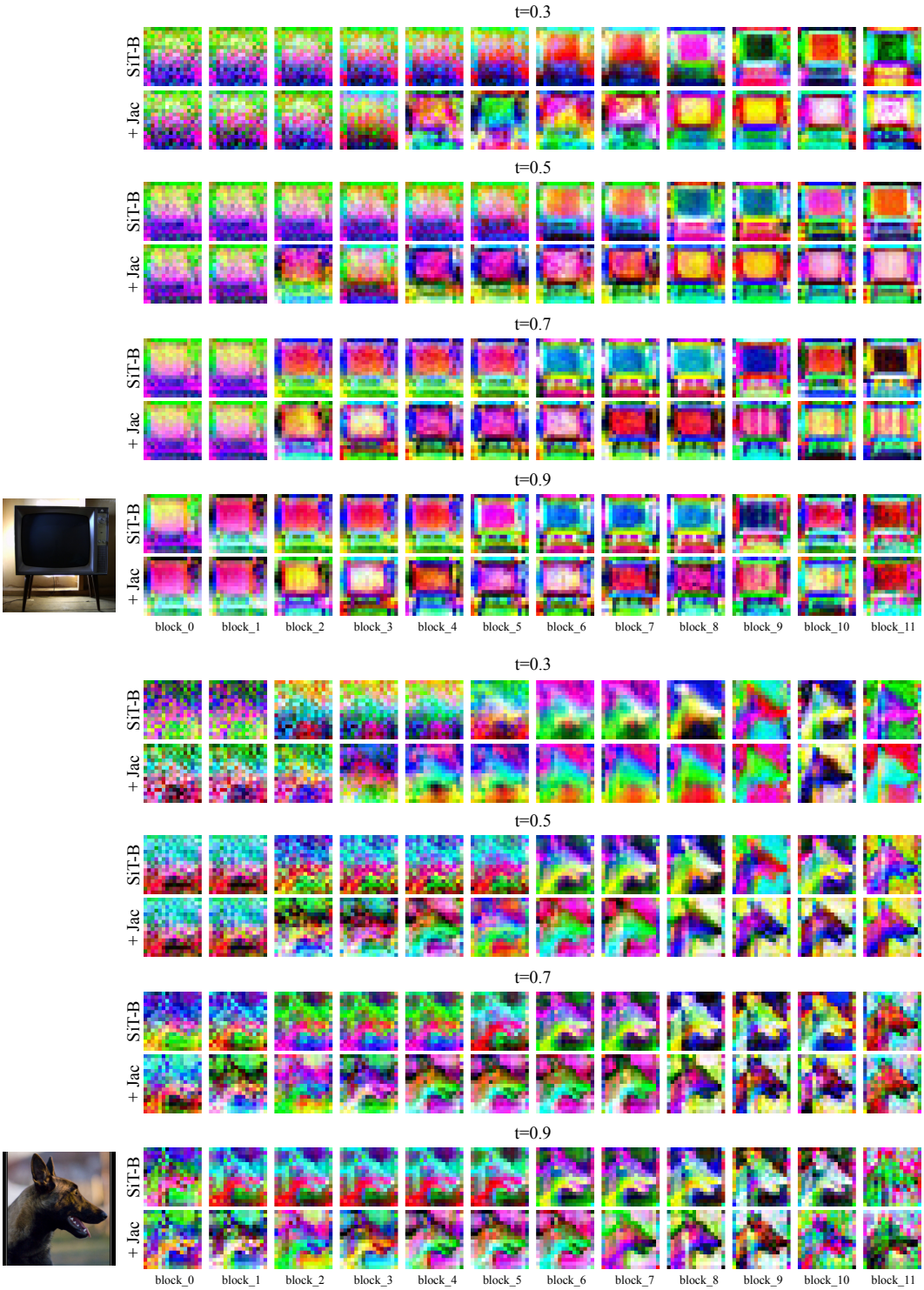


Figure 20: Feature comparison (reduced to RGB using PCA) between the baseline and the Jacobian-regularized SiT-B. At low timesteps ($t = \{0.3, 0.5\}$), the image structures emerge in earlier blocks in the Jacobian-regularized model.



Figure 21: Examples of images generated with the baseline and Jacobian-regularized SiT-B models. We use the Euler sampler with 50 inference steps and guidance scale $w = 4.0$.