Balancing Frequencies and Pixels in Flow Matching

Lucas Degeorge[1,2,3*] **Paul Couairon**[1*] **Arijit Ghosh**[1,3*]
Alexei A. Efros[4] **David Picard**[3†] **Vicky Kalogeiton**[1†]

[1]LIX, École Polytechnique, CNRS, IP Paris, France [2]AMIAD
[3]LIGM, École Nationale des Ponts et Chaussées, IP Paris, UGE, CNRS, France [4]UC Berkeley



Figure 1: (Left) Images generated with our XL model at 256 resolution. Beware, 2 impostors (real images from the ImageNet training set) are hiding in these images. Place your bet on which ones they are and check the solution on page 10. (Right) FID vs. training epoch for an XL model. Our $\boldsymbol{fv}$-loss achieves faster convergence than JiT without requiring architectural modifications.

Abstract

Natural images follow a $1/f^2$ spectral distribution: most signal energy lies in the low spatial frequencies, while the perceptually important structures such as textures and edges occupy sparse high-frequency bands. Pixel-space reconstruction objectives, however, treat all spatial errors uniformly, causing low frequencies to dominate the optimization signal and delaying the learning of fine-scale details. In this work, we identify this objective-level spectral imbalance as a key inefficiency in training pixel-space flow models. To address it, we propose a Focal Log-Frequency Loss ($\boldsymbol{f}$-loss), a spectrally balanced objective that equalizes the learning signal across frequencies, emphasizing high-frequency components that are otherwise underrepresented in pixel-space objectives. Building on this, we introduce a simple training strategy that combines frequency and pixel supervision: we first emphasize frequency-domain learning early to capture all frequencies, and then transition to standard pixel-space v-loss for spatial refinement. This balancing mitigates the low-frequency bias of pixel losses and aligns the training signal with the evolving needs of the model. Our approach is conceptually simple, requires no architectural changes, and acts as a drop-in replacement for flow matching losses. Across multiple model scales, it accelerates convergence by up to 40% while consistently improving FID and perceptual fidelity.

1 Introduction

Images admit two complementary representations: the spatial domain, which captures local structure, and the frequency domain, which captures global patterns across scales. Consider an image of a tiger.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Its global shape is defined by smooth, large-scale variations, while the stripes – thin, structured fine patterns – are encoded in high frequencies. Both are essential: the overall silhouette identifies the object, while the fine patterns make it visually realistic. This duality between pixels and frequencies is fundamental to how images are structured and perceived [12, 16, 59].

A key property of natural images is their spectral organization: their power spectrum approximately follows a $1/f^2$ distribution [49, 61], meaning that most signal energy lies in low frequencies describing global structure, while high frequencies contain sparse but structured details. As a result, low-frequency components dominate the signal energy, even though they represent only part of the perceptual content of the image. In fact, perceptual studies consistently show that high-frequency content plays a disproportionately important role in visual quality [11, 29, 35, 62]. This suggests that image quality is highly sensitive to high-frequency components, such as edges and textures, despite their relatively low energy in the $1/f^2$ spectrum.

Motivated by this discrepancy, we introduce a frequency-domain objective that equalizes the training signal across the spectrum. Specifically, we propose a Focal Log-Frequency Loss ($\boldsymbol{f}$-loss), which applies focal weighting to logarithmically compressed spectral errors. In contrast to the $1/f^2$ distribution of natural images, this formulation increases the influence of high-frequency components that are otherwise underrepresented in pixel-space training [34] (see Figure 2a).

Building on this objective, we study its effect on training dynamics in flow matching models. We empirically show that pixel-space losses [34, 65, 70] are dominated by large low-frequency residuals, biasing early optimization. Emphasizing frequency-domain supervision counteracts this effect, enabling the model to capture structured patterns and fine-scale details early in training. We therefore adopt a simple two-stage strategy: we begin with $\boldsymbol{f}$-loss to learn spectral structure, and subsequently rely on the standard pixel $\boldsymbol{v}$-loss [34] to refine spatial consistency. This simple strategy significantly improves both convergence speed and generation quality(see Figure 1 and 10). Importantly, our method acts as a drop-in replacement for standard flow matching objectives without requiring architectural changes [40], and achieves state-of-the-art performance both quantitatively (FID) and qualitatively for pixel-space class-conditional generation.

Our contributions are threefold:

- We identify a fundamental mismatch between the spectral structure of natural images and pixel-space training objectives, showing that standard losses bias optimization toward low-frequency components despite the perceptual importance of high-frequency details;
- We introduce a spectrally balanced objective, the Focal Log-Frequency Loss ($\boldsymbol{f}$-loss), which equalizes the learning of frequencies and enables models to capture both global structure and fine-scale patterns;
- We demonstrate that emphasizing frequencies during training significantly improves optimization efficiency, while accelerating convergence and consistently improving generation quality across model scales, while requiring no architectural modifications.

2 Related Work

Flow Matching and Pixel-Space Synthesis Denoising generative models including diffusion [9, 20, 27, 28, 42, 54–56] and flow matching frameworks [1, 36, 37], typically frame the learning objective as a spatial-domain regression task, predicting either the noise ($\boldsymbol{\epsilon}$-prediction) or velocity field ($\boldsymbol{v}$-prediction). These perform quite well in recent methods operating in compressed pre-trained latent space [4, 5, 10, 32, 39, 44, 45, 48, 66, 71]. However, when working directly on pixel space both $\boldsymbol{v}$-prediction and $\boldsymbol{\epsilon}$-prediction fail [34]. This is mainly due to the model's under-compute regime, which makes it difficult to separate the full-ranked noise components in its $\boldsymbol{v}$- or $\boldsymbol{\epsilon}$-prediction. Interestingly, natural data lie on a low-dimensional manifold [3], as such, a direct denoised clean data prediction ($\boldsymbol{x}$-prediction) offers the best alternative [34]. This opens the door to adapt architectural modifications [6, 7, 21, 40, 63, 69] as well as additional loss terms [33, 41] for performance gains.

In this work, we look into the main training objective of such pixel space models without any architectural modifications and auxiliary losses and find that a frequency rebalancing loss helps in faster convergence and improved performance.

Spectral Bias and Frequency-Aware Generative Modeling Neural networks inherently exhibit a spectral bias, learning low-frequency structural components significantly faster than fine-grained, high-frequency details [46, 67]. Across generative paradigms, this bias yields models that excel at synthesizing global structure but struggle to preserve sharp textures [26, 48], often resulting in severe high-frequency artifacts [52]. To mitigate these limitations, prior works have explored various interventions. Some approaches balance frequency components post-hoc during sampling [22, 53]. Other works utilize temporal cascading to explicitly decouple frequencies across denoising timesteps [6, 60]. More recently, researchers have introduced structural modifications like multi-scale learning strategies [8, 14, 24, 25], dedicated frequency branches [47, 68], and explicit multi-scale or latent frequency decoupling within the architecture itself [23, 40, 64].

While these methods underscore the critical importance of frequency balancing, they rely on complex denoising schedules, post-hoc adjustments, or complex architectural changes. In contrast, our work targets the fundamental optimization dynamics. We directly correct the spectral bias during training without requiring any architectural modifications.

3 Method

3.1 Background

Flow Matching Models [1, 36, 37] define a regression objective to learn a velocity field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ that transports noise $\boldsymbol{x}_0 \sim p_0$ to data $\boldsymbol{x}_1 \sim p_1$ via the ODE $\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{v}_\theta$. By employing a linear interpolant $\boldsymbol{x}_t = (1-t)\,\boldsymbol{\epsilon} + t\,\boldsymbol{x}_1$, the model can be trained to directly predict the constant ground-truth velocity $\boldsymbol{v}^* = \boldsymbol{x}_1 - \boldsymbol{\epsilon}$. This approach has been widely adopted by state-of-the-art architectures [10, 39] for its stability:

$$\mathcal{L}_{\boldsymbol{v}} = \mathbb{E}\|\boldsymbol{v}_\theta - \boldsymbol{v}^*\|^2 . \tag{1}$$

Prediction Parameterization The choice of prediction target is significantly influenced by the dimensionality of the data. While $\boldsymbol{v}$-prediction is highly effective within compressed latent [10, 31], it often suffers from catastrophic failure when applied to high-dimensional pixel spaces [34]. To address this, the objective is reparameterized to focus the model on $\boldsymbol{x}$-prediction rather than direct velocity regression. Under this framework, the ground-truth velocity is reformulated as: $\boldsymbol{v} = \boldsymbol{x}_1 - \boldsymbol{\epsilon} = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_t}{1-t}$ while $\boldsymbol{v}_\theta = \frac{\boldsymbol{x}_\theta - \boldsymbol{x}_t}{1-t}$. Consequently, the flow matching velocity loss in Equation 1 can be reformulated as a weighted x-regression loss, which we call $\boldsymbol{v}$-loss:

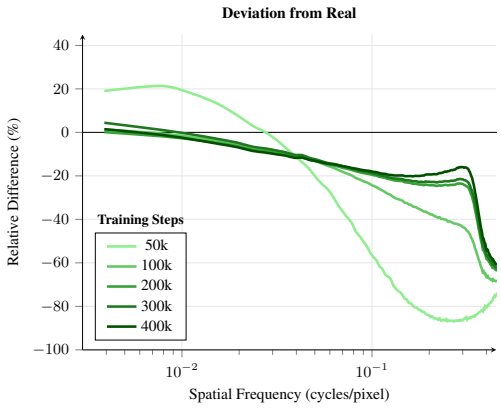
$$\mathcal{L}_{\boldsymbol{v}} = \mathbb{E}\left[\frac{1}{(1-t)^2}\|\boldsymbol{x}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_1\|^2\right] \tag{2}$$

By shifting to $\boldsymbol{x}$-regression, the model is tasked with projecting directly onto the data manifold, a signal that is significantly more tractable than regressing high-variance velocity fields across the high-dimensional ambient space. This better aligns with the model's capacity by focusing the learning process on recovering low dimensional geometric structure rather than modeling high-dimensional temporal transitions.

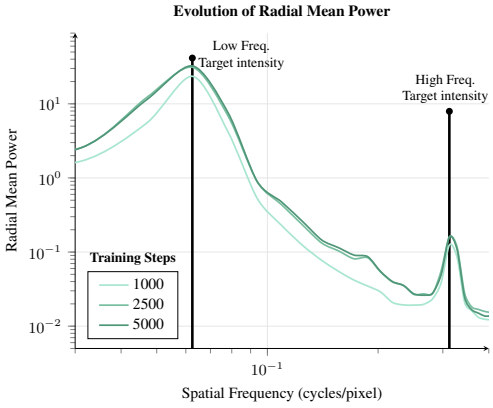
3.2 Spectral Diagnosis of Pixel-Space Flow Models

Neural networks are known to exhibit a spectral bias: they learn low-frequency components of a target function significantly faster than high-frequency ones, with the learning speed at frequency k decaying approximately as $1/k$. Prior works [46, 67] have studied theoretically the bias in the supervised learning setting. We investigate this question empirically through two complementary experiments: a quantitative analysis of the frequency signature of a fully trained model with $\boldsymbol{v}$-loss, and a controlled toy experiment that tracks spectral learning dynamics over the course of training.

Frequency signature of $\boldsymbol{v}$-loss models To directly measure the spectral distortion induced by $\boldsymbol{v}$-loss, we compute the radially averaged power spectrum of images generated by a model trained with $\boldsymbol{v}$-loss, and compare it to the power spectrum of real ImageNet images. Figure 2a reports the relative difference at each spatial frequency. $\boldsymbol{v}$-loss model overestimates by 20% low and mid frequencies (below 10^{-1} cycles/pixel). On the contrary, high frequencies (above 8×10^{-1} cycles/pixel) are underestimated, with the deficit growing toward the Nyquist limit and reaching nearly -60%.



(a) Relative deviation of the generated image power spectrum from real images throughout training. $\boldsymbol{v}$-loss overestimates low frequencies and underestimates high frequencies.



(b) Spectral evolution of a MLP trained with $\boldsymbol{v}$-loss on a two-frequency toy signal. The low-frequency pattern (left) is learned rapidly; the high-frequency pattern (right) remains absent throughout training.

Figure 2: **Spectral bias induced by $\boldsymbol{v}$-loss**: (a) Low frequencies are overrepresented in the radial power spectrum of generated images, (b) High frequencies not learned as well as low frequencies on a simple toy experiment with only 2 active frequencies.

Spectral learning dynamics To isolate the temporal dynamics of this bias, we train a small MLP to generate synthetic images composed of exactly two frequency components: one low and one high. We track the Fourier spectrum of the generated images throughout training. As shown in Figure 2b, the model rapidly learns the low-frequency pattern but struggles to learn the high-frequency one. This result confirms that $\boldsymbol{v}$-loss imposes an implicit preference that puts more emphasis on low frequencies throughout training, making it slow to learn high frequencies associated with fine grain textures. Similarly, in Figure 2a, the model manages to recover the right amount of low frequencies at the end of training, but is still missing some high frequencies. This suggests that by focusing too much on the low frequencies during early training, the capacity of the model to generate high frequencies is irremediably lost and cannot be recovered at later stages.

3.3 A Joint Pixel-Frequency Objective

As mentioned in Section 3.2, supervising only pixel-space residuals creates a systematic spectral imbalance that persists throughout training. This motivates a training objective that operates on both domains jointly: pixel-space supervision to ensure spatial precision, and frequency-space supervision to ensure spectral balance.

Frequency-domain component To ensure such spectral balance, we introduce $\boldsymbol{f}$-loss, a frequency-domain reconstruction objective that reweights the learning signal across the spectral hierarchy:

$$\mathcal{L}_{\boldsymbol{f}} = \sum_{u,v} \frac{1}{(1-t)^2} \cdot \frac{e_{u,v}}{\max\limits_{u',v'} e_{u',v'}} \cdot \log(1 + |\mathcal{F}_{\text{pred}}(u,v) - \mathcal{F}_{\text{target}}(u,v)|) \tag{3}$$

where $\mathcal{F}$ denotes the 2D Discrete Fourier Transform and $e_{u,v} = |\mathcal{F}_{\text{pred}}(u,v) - \mathcal{F}_{\text{target}}(u,v)|$ is the per-frequency residual.

The three components each serve a distinct role. First, the **timestep weighting** $\frac{1}{(1-t)^2}$ is inherited from $\boldsymbol{v}$-loss and preserved unchanged, ensuring consistent timesteps dynamics. Then, the **adaptive focal weight** $\frac{e_{u,v}}{\max\limits_{u',v'} e_{u',v'}}$ normalizes the per-frequency residuals by the maximum residual across the spectrum at each training step. This is computed with a stop-gradient and acts as a purely data-dependent coefficient: it continuously normalizes each sample to avoid wide variations in magnitude between samples. Finally, the **logarithmic compression** $\log(1 + e_{u,v})$ prevents any single frequency from monopolizing the loss, and can be interpreted as a continuous generalization of Laplacian

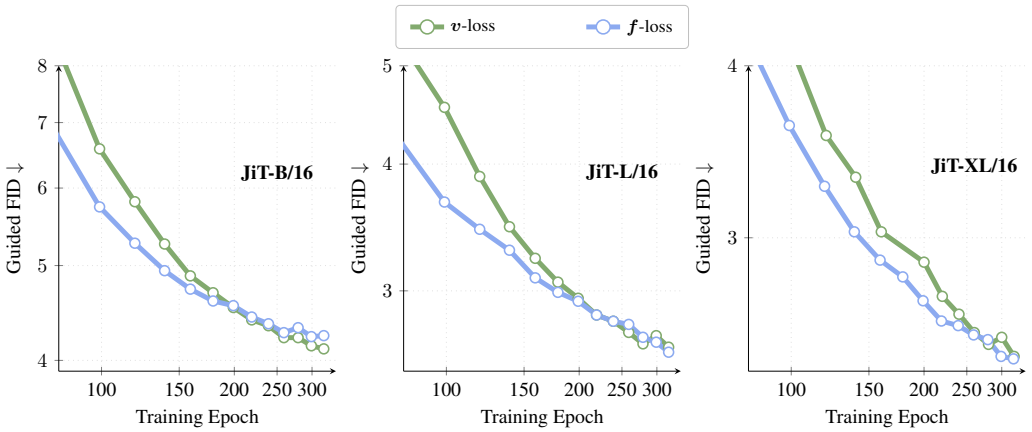


Figure 3: **Convergence of frequency- and pixel-space losses.** The frequency-domain loss dominates early in training, while the spatial-domain loss gradually catches up in later stages.

pyramid decomposition: just as the pyramid assigns equal structural importance to each octave, the logarithm linearizes the spectral hierarchy, granting every doubling of frequency equal weight in the total loss (see Appendix D).

Joint objective We combine the frequency-domain component with the standard pixel component $\boldsymbol{v}$-loss into a single joint objective:

$$\mathcal{L} = w_f \cdot \mathcal{L}_f + w_v \cdot \mathcal{L}_v \tag{4}$$

where w_f and w_v are scalars that control the relative contribution of each domain throughout training. Empirically, we observe in Figure 3 that $\boldsymbol{f}$-loss dominates early in training, converging significantly faster than $\boldsymbol{v}$-loss. However, $\boldsymbol{v}$-loss surpasses $\boldsymbol{f}$-loss later in training. It suggests that the two components are not equally useful at all stages: frequency supervision is the binding constraint early on, while pixel supervision becomes more effective as training progresses. We set $w_f = \lambda(s)$ and $w_v = 1 - \lambda(s)$ where $\lambda(s)$ follows a sigmoid schedule decaying from 1 to 0, centered at the observed crossover point $s^\star$. The design choices for the combination schedule are ablated in Section 4.2. We call the resulting scheduled loss $\boldsymbol{fv}$-loss.

3.4 Empirical Validation

Convergence speedup Figure 4 reports FID across training epochs for JiT-B/16, JiT-L/16, and JiT-XL/16, in both unguided and guided settings. The joint $\boldsymbol{fv}$-loss objective outperforms the baseline $\boldsymbol{v}$-loss across all model sizes from the earliest evaluations. The gap is largest in the early epochs and narrows as training progresses, consistent with our observation that frequency supervision is most valuable early in training.

Qualitative evidence Figure 5 compares images generated by $\boldsymbol{v}$-loss and $\boldsymbol{f}$-loss at two stages of training: very early (left block) and early (right block). The difference is striking at very early training stages. $\boldsymbol{f}$-loss produces images with significantly finer textures well before $\boldsymbol{v}$-loss: the hay on the thatched roof is detailed, and the knitting pattern on the mittens is already visible. $\boldsymbol{v}$-loss at the same stage appears to prioritize sharp object boundaries over surface texture. Later in training, $\boldsymbol{v}$-loss recovers many of these textures, though they remain less defined than those produced by $\boldsymbol{f}$-loss.

Frequency-Domain Error Analysis Figure 6 (right) measures the average magnitude error between generated and real images in the low- and high-frequency bands of the Fourier spectrum. In the earliest stages of training (50k-75k steps), $\boldsymbol{f}$-loss achieves a lower error than $\boldsymbol{v}$-loss in both bands, confirming that it corrects the spectral bias faster. As training progresses, $\boldsymbol{f}$-loss plateaus and is eventually overtaken by $\boldsymbol{v}$-loss. $\boldsymbol{fv}$-loss combines the fast early spectral alignment of $\boldsymbol{f}$-loss with the late-stage refinement of $\boldsymbol{v}$-loss, reaching the lowest error in both bands by the end of training.

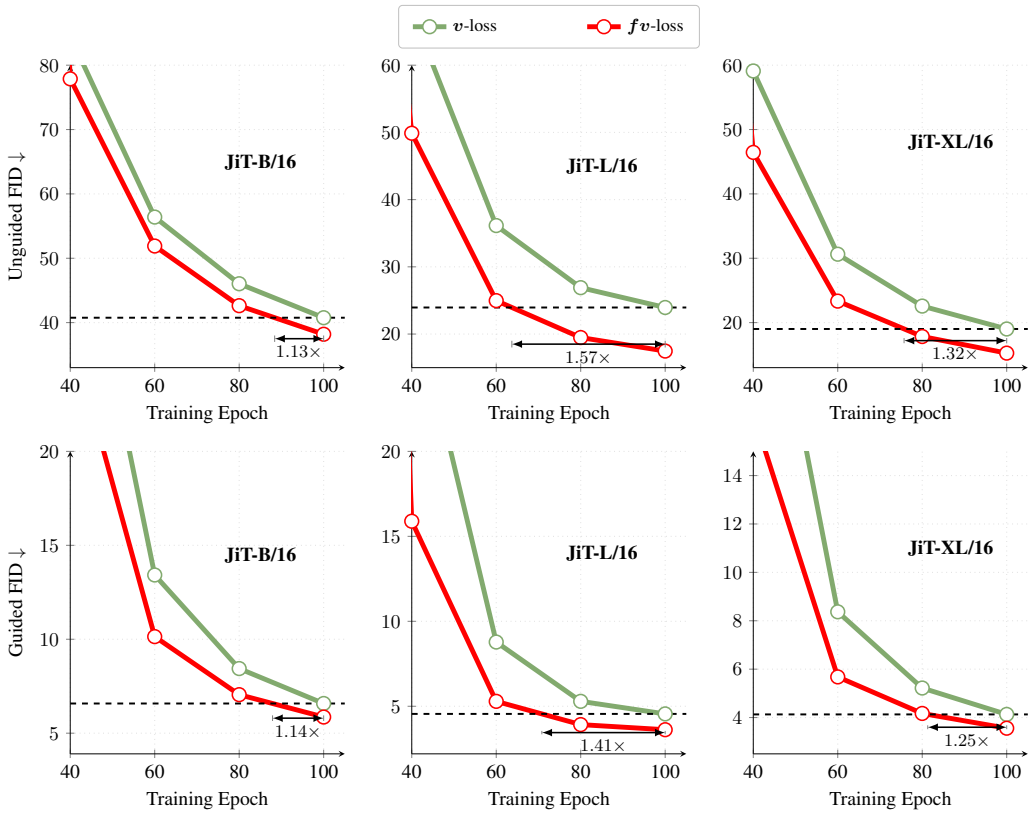


Figure 4: **Convergence speed comparison** between $\boldsymbol{v}$-loss and $\boldsymbol{fv}$-loss for different model sizes. Top row: unguided FID. Bottom row: guided FID. $\boldsymbol{fv}$-loss consistently outperforms $\boldsymbol{v}$-loss leading to significant speed-ups.

f-loss forces the model to escape this low-frequency bias by actively equalizing the learning signal across all frequency bands. But it is insensitive to phase, which carries the precise spatial location of edges and textures. $\boldsymbol{v}$-loss directly supervises pixel values and is ultimately better suited for enforcing strict local spatial coherence and phase alignment (e.g., locking sharp edges into exact pixel locations). We hypothesize that this fundamental difference explains why the f-loss eventually plateaus when used alone, making the transition back to the spatial-domain v-loss necessary for final refinement.

4 Experiments

In this section, we discuss the effectiveness of training an image generative model using our proposed $\boldsymbol{fv}$-loss. We detail our experimental setup and our evaluation metric to support our experiments. Based on these, we see how our proposed method performs against state-of-the-art models in Section 4.1. Following that, we show empirical analysis for specific components of our proposed $\boldsymbol{fv}$-loss and compare it against the standard $\boldsymbol{v}$-loss to study the convergence dynamics in Section 4.2.

Experimental setup. For fair and reproducible evaluations, we train our models using the ImageNet class conditional setup [50]; at 256×256 and 512×512 resolutions. We primarily focus on pixel-space flow matching framework following the `JiT` architecture [34] with x-pred and a reformulated loss. Unless we state otherwise, models are trained using identical training schedules and compute budgets (batch size: 1024; lr: constant at 2e-4 after 5 epochs of warmup; optim: AdamW [38] with β_1, β_2: 0.9, 0.95), allowing us to isolate the effect of the proposed loss. For evaluation, we generate 50K samples using 50 ODE steps of Heun sampler [17] and compute all metrics following the standard protocols used in recent generative modeling benchmarks [9]. We refer readers to Table 5 for more detailed information about experimental hyperparameters.

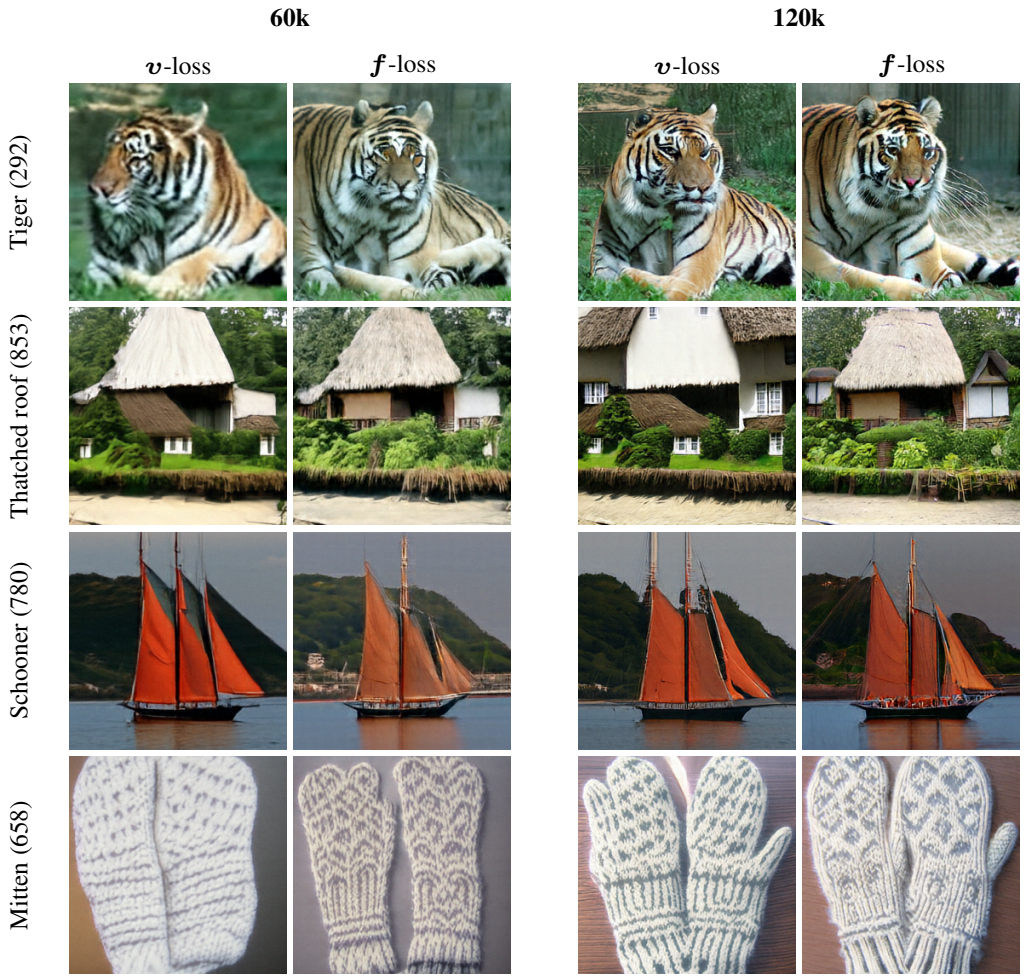


Figure 5: **Generated images depending on the loss at early training stages.** (left) f-loss images have finer textures (*e.g.*, the tiger's stripes, the hay on the thatched roof) much earlier than v-loss that seems to focus more on sharp object boundaries (*e.g.*, well defined sails of the boat) at such early stages. (right) These finer textures starts to appear with v-loss later during training, which may explain its slower start in FID. Please refer to Figure 9 for comparison with finer number of steps.

Evaluation Metrics. To evaluate the effectiveness of our proposed method, we primarily use the standard **Fréchet Inception Distance** (`FID`) [18] to quantify the quality of the generated samples. `FID` measures distance between the `InceptionV3` [58] feature distributions of generated and real images, providing a standard measure of overall generation quality. Additionally, for our ablations we also evaluate using **Fréchet Dino Distance** (`FDD`) [57], which calculates the Fréchet distance but based on `DinoV2` [43] features. Finally, following prior works [9, 34], we also evaluate using **Inception Score** (`IS`) [51] to show that our proposed method does pick up on class-specific information.

4.1 State-of-the-art comparison

We compare our proposed fv-loss against the state-of-the-art pixel flow models in Table 1. Our proposed fv-loss improves upon the standard `JiT` framework w.r.t. the same model architecture (`XL/16`) and training settings (2.13 vs 2.21 after 750K steps). Following recent methods [6, 40] we also apply `REPA` alignment loss. Interestingly, with our proposed method we match `DeCo` performance in less than half the training steps. Finally, when adding perceptual losses similar to `PixelGen`, our proposed method improves on the current state-of-the-art performance and achieves the same FID of 1.83 in only 500k steps (compared to 800k for PixelGen) while significantly improving on IS, without any additional architectural changes.

Method	Model size	REPA	Perceptual	Training steps	FID↓	IS↑
JiT-H/16 [34]	H/16	✗	✗	750k	1.86	303.4
JiT-G/16 [34]	G/16	✗	✗	750k	1.82	292.6
JiT-XL/16 [34]	XL/16	✗	✗	750k	2.21	**297.4**
Ours	XL/16	✗	✗	750k	**2.13**	290.3
EPG [33] ICLR, 2026	XL/16	✓	✗	1M	2.04	283.2
PixelFlow [6] 2025	XL/4	✓	✗	1.6M	1.98	282.1
DiP [7] CVPR, 2026	XL/16	✓	✗	1.6M	1.98	282.9
PixNerd [63] ICLR, 2026	XL/16	✓	✗	1.6M	1.95	298.0
DeCo [40] CVPR, 2026	XL/16	✓	✗	1.6M	1.90	**303.0**
Ours	XL/16	✓	✗	750k	**1.87**	301.0
PixelGen [41] 2026	XL/16	✓	✓	800k	**1.83**	293.6
Ours	XL/16	✓	✓	500k	**1.83**	**323.4**

Table 1: **Comparison with state-of-the-art pixel-space methods.** Our proposed loss is able to improve over all categories of models (standard, using alignment methods like REPA, using perceptual losses), either by achieving strictly better FIDs or by reaching the same FID faster.

	Epochs	Loss	FID↓		FDD↓	IS↑
			w/o cfg	cfg	cfg	cfg
JiT-B/16	80	v-loss	46.02	8.44	369.3	162.8
		f-loss	**42.34**	**6.82**	**312.0**	**186.6**
		fv-loss	42.63	7.05	314.0	185.2
	200	v-loss	31.23	4.52	219.8	252.1
		f-loss	**30.02**	4.54	**194.8**	254.3
		fv-loss	30.17	**4.20**	202.3	**258.9**
	320	v-loss	28.70	4.11	192.2	273.9
		f-loss	27.51	4.23	**170.4**	275.2
		fv-loss	**26.86**	**3.95**	178.8	**282.6**
JiT-L/16	80	v-loss	26.90	5.40	264.2	200.0
		f-loss	20.24	4.34	185.8	237.3
		fv-loss	**19.47**	**3.93**	**178.6**	**247.9**
	200	v-loss	16.54	2.98	155.4	274.3
		f-loss	**14.14**	2.92	125.1	283.3
		fv-loss	14.79	**2.82**	**122.2**	**291.6**
	320	v-loss	14.46	2.63	133.7	292.7
		f-loss	13.26	2.58	115.2	287.5
		fv-loss	**12.93**	**2.55**	**113.7**	**303.4**

(a) Resolution 256×256

	Epochs	Loss	FID↓		FDD↓	IS↑
			w/o cfg	cfg	cfg	cfg
JiT-B/32	80	v-loss	52.72	10.40	833.8	153.7
		f-loss	50.54	9.48	792.3	159.0
		fv-loss	**45.90**	**8.33**	**775.1**	**175.2**
	200	v-loss	35.20	5.18	676.4	244.5
		f-loss	36.54	5.19	664.1	231.9
		fv-loss	**33.12**	**4.74**	**653.7**	**254.3**
	320	v-loss	31.94	4.63	642.8	270.1
		f-loss	32.93	4.64	634.5	254.9
		fv-loss	**29.12**	**4.47**	**626.9**	**281.9**
JiT-L/32	80	v-loss	29.04	6.41	746.4	228.2
		f-loss	22.91	4.65	689.5	**237.0**
		fv-loss	**21.64**	**4.58**	**681.9**	196.9
	200	v-loss	17.80	3.22	650.6	**288.8**
		f-loss	16.71	3.26	633.3	279.1
		fv-loss	**15.84**	**2.98**	**631.0**	284.5
	320	v-loss	15.85	2.90	639.8	294.6
		f-loss	15.28	3.04	**619.5**	282.4
		fv-loss	**14.30**	**2.73**	620.2	**312.1**

(b) Resolution 512×512

Table 2: **Comparison between v-loss, f-loss and fv-loss** at resolutions (a) 256×256 and (b) 512×512. fv-loss combines the best of v-loss and f-loss, namely speed of convergence at early stages and final performances.

4.2 Ablation Study

We begin by first ablating our proposed $\boldsymbol{fv}$-loss (given in Equation 4). In the 256^2 resolution setup for a B size model (see Table 2a), we see that using simple $\boldsymbol{f}$-loss over $\boldsymbol{v}$-loss is highly beneficial early in the training for both unguided (42.34 vs 46.02) and guided (6.82 vs 8.44) settings. As training progresses, the $\boldsymbol{f}$-loss is still better than $\boldsymbol{v}$-loss w.r.t unguided setup (27.51 vs 28.70). A similar trend can also be seen for bigger L size models too, where $\boldsymbol{f}$-loss is better than the $\boldsymbol{v}$-loss early in the training for unguided settings (20.24 vs 26.90 at 80 epochs) and remains on par at the end (13.26 vs 14.46 at 320 epochs). However, for the guided setup, the $\boldsymbol{v}$-loss becomes better than the $\boldsymbol{f}$-loss at later

w_f	w_v	FID ↓			IS ↑
		Epoch 80	Epoch 200	Epoch 320	
1	0	6.82	4.54	4.23	275.18
0	1	8.44	4.52	4.11	273.85
	0.1	7.24	4.45	4.16	269.16
	0.5	6.85	4.49	4.31	279.18
1	1	7.04	4.33	4.05	274.85
	5	7.05	4.51	4.22	282.47
	10	7.14	4.25	4.02	**285.88**
$1-\lambda(s)$	$\lambda(s)$	8.52	4.48	4.30	265.9
$\lambda(s)$	$1-\lambda(s)$	7.05	4.20	**3.95**	282.58

Table 3: Ablation on the weights w_f and w_v of $\boldsymbol{fv}$-loss

training stage, irrespective of the model size. We argue that this is because, as training progresses, it is more important for the model to learn from the pixels to improve upon the high-frequency details. Interestingly, with this in mind our proposed $\boldsymbol{fv}$-loss setup with a switch from frequency to spatial domain shows improvement both in the early stages of training as well as in the later stages for both unguided and guided setups.

A similar trend can also be observed for 512^2 resolution experiments (see Table 2b). In the early stages of training $\boldsymbol{f}$-loss is always better than $\boldsymbol{v}$-loss irrespective of model sizes (for `B`: 9.48 vs 10.40; for `L`: 4.65 vs 6.41 at 80 epochs). However, with training progression, a switch using $\boldsymbol{fv}$-loss is much better compared to $\boldsymbol{v}$-loss and $\boldsymbol{f}$-loss alone.

This provides empirical evidence that, indeed, beginning with a much more balanced frequency-based objective is beneficial for better pixel space models. However, at the end, where there is much more need for fine-grained details, reverting back to a spatial loss is highly beneficial.

Adaptative weighting To understand the switch in $\boldsymbol{f}$-loss as given in Equation 4, we also ablate the effect of w_f and w_v. As reported in Table 3, clearly having a sigmoid weight performs the best throughout training, w.r.t. `FID`. Interestingly, a higher weight on $\boldsymbol{v}$-loss also helps but is not as good as the proposed sigmoid switch. Crucially, a reversed sigmoid that emphasizes mostly pixel loss at the beginning and frequencies loss at later stage has among the worst performances. This clearly indicates that a stronger pixel flow model requires some spectral balancing in the early training stage to set the stage for spatial high-detail learning.

Generalization across architectures. To test whether the benefits of $\boldsymbol{fv}$-loss generalize beyond JiT, we train PixelDiT [69] with v-loss and $\boldsymbol{fv}$-loss following the original PixelDiT setup. Table 4 shows that $\boldsymbol{fv}$-loss reaches the same FID roughly $2\times$ faster than v-loss. Moreover, PixelDiT is trained with the REPA alignment loss. REPA pushed the features toward rich high-frequencies DINO features. $\boldsymbol{fv}$-loss is beneficial even when used alongside REPA.

	Loss	Step 200k	Step 400k	Step 600k
PixelDiT	v-loss (reported from [69])	15	12	8
	v-loss (reproduced)	15.09	10.60	10.25
	$\boldsymbol{fv}$-loss	**10.18**	**6.88**	**5.92**

Table 4: **FID Comparison of v-loss and $\boldsymbol{fv}$-loss using PixelDiT [69] architecture.**

Wall-clock speedup. Figure 6 reports the guided FID reached by each loss at fixed wall-clock budgets, using the training step closest to each time mark. As $\boldsymbol{f}$-loss requires computing a 2D FFT at every training step, it introduces a computational overhead. To verify that this does not negate the reported convergence speed-up, we measure the total wall-clock time for training JiT-B for $400k$ steps. Compared to the $\boldsymbol{v}$-loss baseline (432.0min), $\boldsymbol{f}$-loss incurs a $+4\%$ overhead (453.1min). $\boldsymbol{fv}$-loss

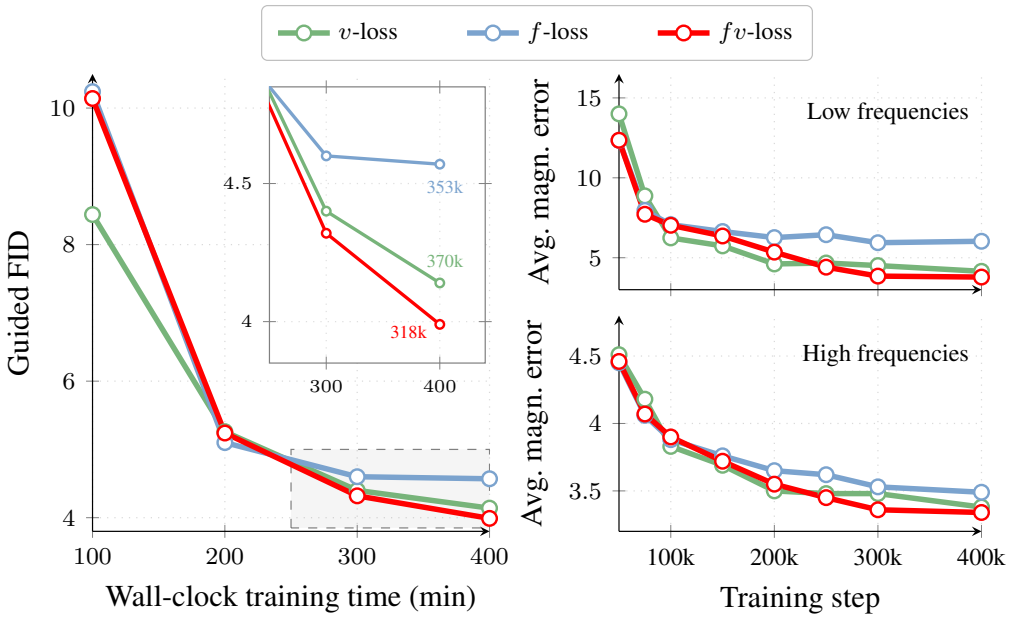


Figure 6: **Left:** Guided FID against wall-clock training time; the number of training steps is given for the last point of each curve. **Right:** Average magnitude error in the low- and high-frequency bands. $\boldsymbol{fv}$-loss consistently outperforms $\boldsymbol{v}$-loss, leading to significant speed-ups.

exhibits a $+14\%$ overhead (492.3min), as it computes and backpropagates both losses at each step. At the 100-minute mark, $\boldsymbol{v}$-loss is temporarily ahead because $\boldsymbol{fv}$-loss has completed fewer steps ($90k$ vs. $78k$ steps). However, $\boldsymbol{fv}$-loss overtakes $\boldsymbol{v}$-loss by the 200-minute mark. From 300 minutes onward, its convergence speed-up more than compensates for the added computational cost.

5 Conclusion

In this paper, we investigate spectral bias in flow-matching models and show that the pixel-space velocity loss of JiT leads to overestimating low frequencies early during training, at the cost of underestimating higher frequencies. This can be explained by the natural $1/f^2$ decay of the power spectrum of natural images that leads to imbalance in the training objective. This bias towards low frequencies at early training stages seems to be hurting early convergence. As such, we propose a new loss in the frequency domain that aims at mitigating such bias by equalizing frequencies. We experimentally show that such loss obtains better performances at early training stages. We propose to combine the early speed advantage of the frequency loss with the later training performances of the classical pixel-space velocity loss through a schedule. The resulting $\boldsymbol{fv}$-loss significantly accelerate the convergence of flow-matching training and leads to state-of-the-art pixel-space image generation on class-conditional ImageNet.

Limitations Whereas the spectral bias of modern pixel space flow-matching models is clearly exposed and a mitigation strategy is successfully proposed, understanding its origin remains an open question. More specifically, future work should investigate why having a frequency objective at early training stages plateaus in performances when used alone, and why switching to pixel space losses performs better at later training stages.

Acknowledgments and Disclosure of Funding

This work was supported by a Hi!Paris grant, two Hi!Paris chaires for A.A.Efros and V.Kalogeiton, ANR/France 2030 program (ANR-23-IACL-0005) and ANR project sharp ANR-23- PEIA-0008 in the context of the PEPR IA. It was granted access to the HPC resources of IDRIS under the allocations

The real images in the teaser are the tiger and the magpie.

2025-A0181016194, 2025-AD011015436, 2026-A0201017545 and 2026-AD011015594R2 made by GENCI. We would like to thank Nicolas Dufour, Julie Mordacq, Robin Courant, Yohann Perron, Adrien Ramanana Rahary and Hovhannes Margaryan for their helpful comments and proofreading.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2022. 2, 3

[2] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851. 18

[3] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006. 2

[4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *ICLR*, 2023. 2

[5] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *ECCV*, pages 74–91. Springer, 2024. 2

[6] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025. 2, 3, 7, 8

[7] Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space. *arXiv:2511.18822*, 2025. 2, 8

[8] Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *NeurIPS*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.NEURIPS.cc/paper_files/paper/2015/file/aa169b49b583a2b5af89203c2b78c67c-Paper.pdf. 3

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 2, 6, 7

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B. 2, 3

[11] Rony Ferzli and Lina J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *TIP*, 2009. 2

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2018. 2

[13] 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. *arXiv:1706.02677*, 2017. 16

[14] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua M. Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *ICLR*, 2024. URL https://openreview.net/forum?id=tOzCcDdH9O. 3

[15] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, 1995. 18

[16] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. In *NeurIPS*, 2020. 2

[17] Karl Heun et al. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45(23-38):7, 1900. 6, 16

[18] 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. *NeurIPS*, 2017. 7

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 16

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2

[21] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In *CVPR*, pages 18062–18071, 2025. 2

[22] Linjiang Huang, Rongyao Fang, Aiping Zhang, Guanglu Song, Si Liu, Yu Liu, and Hongsheng Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. In *European conference on computer vision*, pages 196–212. Springer, 2024. 3

[23] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *ICCV*, pages 13919–13929, 2021. 3, 16, 17, 18

[24] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *CVPR*, pages 10124–10134, 2023. 3

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hk99zCeAb`. 3

[26] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *NeurIPS*, 34:852–863, 2021. 3

[27] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 2

[28] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024. 2

[29] Vishwakumara Kayargadde and Jean-Bernard Martens. Perceptual characterization of images degraded by blur and noise: experiments. *J. Opt. Soc. Am. A*, 1996. 2

[30] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *NeurIPS*, 2024. 16

[31] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`. 3

[32] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2

[33] Jiachen Lei, Keli Liu, Julius Berner, Y HoiM, Hongkai Zheng, Jiahong Wu, and Xiangxiang Chu. There is no VAE: End-to-end pixel-space generative modeling via self-supervised pre-training. In *ICLR*, 2026. URL `https://openreview.net/forum?id=HbUoKPIZmp`. 2, 8

[34] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. 2, 3, 6, 7, 8, 16, 27

[35] Justin D Lieber, Gerick M Lee, Najib J Majaj, and J Anthony Movshon. Naturalistic texture perception relies preferentially on high spatial frequencies. *Journal of Vision*, 2020. 2

[36] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2022. 2, 3

[37] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2022. 2, 3

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 16

[39] 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*, pages 23–40. Springer, 2024. 2, 3, 17, 19

[40] Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. *arXiv preprint arXiv:2511.19365*, 2025. 2, 3, 7, 8

[41] Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026. 2, 8

[42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. Proceedings of Machine Learning Research, 2021. 2

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 7

[44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. URL https://openreview.net/forum?id=di52zR8xgf. 2

[46] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *ICML*, pages 5301–5310. PMLR, 2019. 3

[47] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Frequency-aware flow matching for high-quality image generation, 2026. URL https://arxiv.org/abs/2604.15521. 3

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 17

[49] Daniel Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993. URL https://proceedings.neurips.cc/paper_files/paper/1993/file/4d5b995358e7798bc7e9d9db83c612a5-Paper.pdf. 2

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 6, 27

[51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 7

[52] Katja Schwarz, Yiyi Liao, and Andreas Geiger. On the frequency bias of generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[53] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *CVPR*, 2024. 3

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015. 2

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. URL `https://openreview.net/forum?id=St1giarCHLP`.

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2020. 2

[57] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023. 7

[58] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[59] Alexa R Tartaglini, Wai Keen Vong, and Brenden M Lake. A developmentally-inspired examination of shape versus texture bias in machines. *arXiv preprint arXiv:2202.08340*, 2022. 2

[60] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *ICLR*, 2024. 3

[61] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: computation in neural systems*, 14(3):391, 2003. 2

[62] Ewout Vansteenkiste, Dietrich Van der Weken, Wilfried Philips, and Etienne Kerre. Perceived image quality measurement of state-of-the-art noise reduction schemes. In *Advanced Concepts for Intelligent Vision Systems*, 2006. 2

[63] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=BDnOrExHmt`. 2, 8

[64] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *CVPR*, 2026. 3

[65] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. 2

[66] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 2

[67] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pages 264–274. Springer, 2019. 3

[68] Sunjae Yoon, Gwanhyeong Koo, Geonwoo Kim, and Chang D Yoo. Frag: Frequency adapting group for diffusion video editing. In *ICML*, 2024. 3

[69] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025. 2, 9

[70] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3:47–57, 2017. URL `https://api.semanticscholar.org/CorpusID:5334482`. 2

[71] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026. URL `https://openreview.net/forum?id=0u1LigJaab`. 2

A Implementation Details

Our implementation closely follows the public codebases of JiT [34]. Table 5 details the architecture, training, and sampling parameters. All experiments were run on 4 nodes of 4 H100 GPUs. The final XL models were trained on 4 nodes of 4 GB200 GPUs.

We select the switch point $s^\star$ of Equation 4 using a simple crossing-point rule: we identify where the $\boldsymbol{f}$-loss and $\boldsymbol{v}$-loss FID curves intersect on a held-out validation set (around 200k steps for the B and L models, and around 350k steps for the XL models), and do not perform a grid search over $s^\star$.

	JiT-B	**JiT-L**	**JiT-XL**
architecture			
depth	12	24	28
hidden dim	768	1024	1152
heads	12	16	16
patch size	`image_size` / 16		
bottleneck	128		
in-context class tokens	32		
in-context start block	4	8	8
training			
warmup epochs [13]	5		
optimizer	AdamW [38], $\beta_1, \beta_2 = 0.9, 0.95$		
batch size	1024		
learning rate	2e-4		
learning rate schedule	constant		
weight decay	0		
ema decay	0.9999		
time sampler	logit$(t) \sim \mathcal{N}(\mu, \sigma^2)$, μ = -0.8, σ = 0.8		
noise scale	1.0 $\times$ `image_size` / 256		
clip of $(1-t)$ in division	0.05		
class token drop (for CFG)	0.1		
sampling			
ODE solver	Heun [17]		
ODE steps	50		
time steps	linear in [0.0, 1.0]		
CFG scale sweep range [19]	[1.0, 4.0]		
CFG interval [30]	[0.1, 1] (if used)		

Table 5: **Configurations of experiments.**

B Additional Results

B.1 Design choice of f-loss and fv-loss

In this section, we ablate the design components of $\boldsymbol{f}$-loss using a B-size model and report the results in Table 6. Note that those ablations have been done with cfg interval, RoPE and in-context tokens. First, we compare against the original focal frequency loss [23] (first row), which performs significantly worse than our formulation. The key difference between the two formulations is the logarithmic compression of the spectral residual: while the loss of [23] reweights the raw magnitude error, $\boldsymbol{f}$-loss compresses it with $\log(1 + e_{u,v})$ before reweighting, which we ground as a continuous generalization of a Laplacian pyramid (Appendix D): it assigns comparable structural importance to each octave instead of letting the largest residuals dominate the loss. Introducing the logarithmic compression of spectral residuals already yields substantial improvements, confirming that the $\log$ transformation (as motivated by the scale hierarchy of Laplacian pyramids) plays a key role in stabilizing the spectral balancing.

We further observe that normalized weighting schemes consistently outperform their unnormalized counterparts. Without normalization, the loss magnitude becomes dominated by a subset of fre-

quencies, preventing effective balancing across the spectrum. In contrast, normalizing the spectral weights ensures that the relative difficulty of different frequencies is preserved during optimization. Combining both components yields the best performance, achieving an FID of 41.78 (cfg=1.0) and 6.39 (cfg=3.5).

Frequency-term and scheduler ablation The frequency-to-pixel schedule of Equation 4 is agnostic to the specific frequency-domain loss used. To test the effectiveness of this schedule, we replace $\boldsymbol{f}$-loss with alternative frequency objectives in the scheduled loss. Table 7 reports the results using JiT-B at 256² resolution. We compare a discrete Focal Frequency Loss (row 2), the original loss of [23] (row 3), and a Laplacian pyramid loss (row 4). While the schedule proves beneficial across all variants by Epoch 320, the alternative frequency losses perform substantially worse than $\boldsymbol{f}$-loss and $\boldsymbol{v}$-loss early in training (Epoch 80), yielding FIDs of 21.22 and 23.93 for the loss of [23] and the Laplacian loss, respectively. Only the discrete Focal Frequency Loss is competitive with $\boldsymbol{f}$-loss early on. However, only the continuous formulation $\boldsymbol{fv}$-loss yields the best FID at every stage of training.

B.2 Frequency-Domain Error Analysis

Figure 6 (right) and Figure 7 (full) measures the average magnitude error between generated and real images in the low- and high-frequency bands of the Fourier spectrum. In the earliest stages of training (50k-75k steps), $\boldsymbol{f}$-loss achieves a lower error than $\boldsymbol{v}$-loss in both bands, confirming that it corrects the spectral bias faster. As training progresses, $\boldsymbol{f}$-loss plateaus and is eventually overtaken by $\boldsymbol{v}$-loss. $\boldsymbol{fv}$-loss combines the fast early spectral alignment of $\boldsymbol{f}$-loss with the late-stage refinement of $\boldsymbol{v}$-loss, reaching the lowest error in both bands by the end of training.

f-loss forces the model to escape this low-frequency bias by actively equalizing the learning signal across all frequency bands. But it is insensitive to phase, which carries the precise spatial location of edges and textures. $\boldsymbol{v}$-loss directly supervises pixel values and is ultimately better suited for enforcing strict local spatial coherence and phase alignment (e.g., locking sharp edges into exact pixel locations). We hypothesize that this fundamental difference explains why the f-loss eventually plateaus when used alone, making the transition back to the spatial-domain v-loss necessary for final refinement.

B.3 Generalization to Latent-Space Models

The spectral bias we address in this work is predominantly a pixel-space phenomenon. When operating in a compressed latent space, the data has already passed through a VAE, which by construction discards much of the high-frequency detail present in pixel space [48], so it is unclear whether – or how – the same bias manifests in latent space. To probe this empirically, we train an SiT-B [39] model with x-prediction (required to compute the Fourier spectrum) using v-loss and $\boldsymbol{f}$-loss. Table 8 shows that $\boldsymbol{f}$-loss still yields a lower FID than v-loss at epoch 80, though neither reaches the performance of the original SiT model trained with v-prediction (FID 33 at epoch 80). This suggests that while the mechanism we identify is chiefly a pixel-space one, some benefit of frequency-domain supervision may transfer to latent-space training, which we leave for future work.

Loss	$w_{u,v}$	$\log(1+e_{u,v})$	**FID** ↓ cfg=1.0	**FID** ↓ cfg=3.5	**Prec.** ↑	**Rec.** ↑	**Den.** ↑	**Cov.** ↑
`focal` [23]	$\frac{e_k}{\max_{k'} e_{k'}}$	✗	72.80	19.24	0.38	0.54	0.32	0.60
$\boldsymbol{f}$-loss	1	✗	50.12	12.67	0.49	0.63	0.48	0.77
	1	✓	57.17	17.01	0.43	0.61	0.39	0.72
	$\frac{\log(1+e_k)}{\max_{k'} \log(1+e'_k)}$	✓	44.40	7.42	0.55	0.65	0.61	0.83
	$\frac{e_k}{\max_{k'} e_{k'}}$	✓	**41.78**	**6.39**	**0.56**	**0.66**	**0.62**	**0.85**

Table 6: Ablation study of the $\boldsymbol{f}$-loss components. (Note: The Precision, Recall, Density and Coverage are provided for unguided settings)

Loss	Epoch 80	Epoch 200	Epoch 320
v-loss	8.44	4.52	4.11
Discrete Focal Frequency Loss $\rightarrow$ v-loss	8.92	4.39	4.13
Focal Loss from [23] $\rightarrow$ v-loss	21.22	6.80	4.72
Laplacian loss $\rightarrow$ v-loss	23.93	5.04	4.12
$\boldsymbol{fv}$-loss	**7.05**	**4.20**	**3.95**

Table 7: **Frequency-to-pixel schedule with alternative frequency losses.** Guided FID (B-size model, 256^2) when the frequency-to-pixel schedule of Equation 4 is applied to different frequency-domain losses instead of $\boldsymbol{f}$-loss.

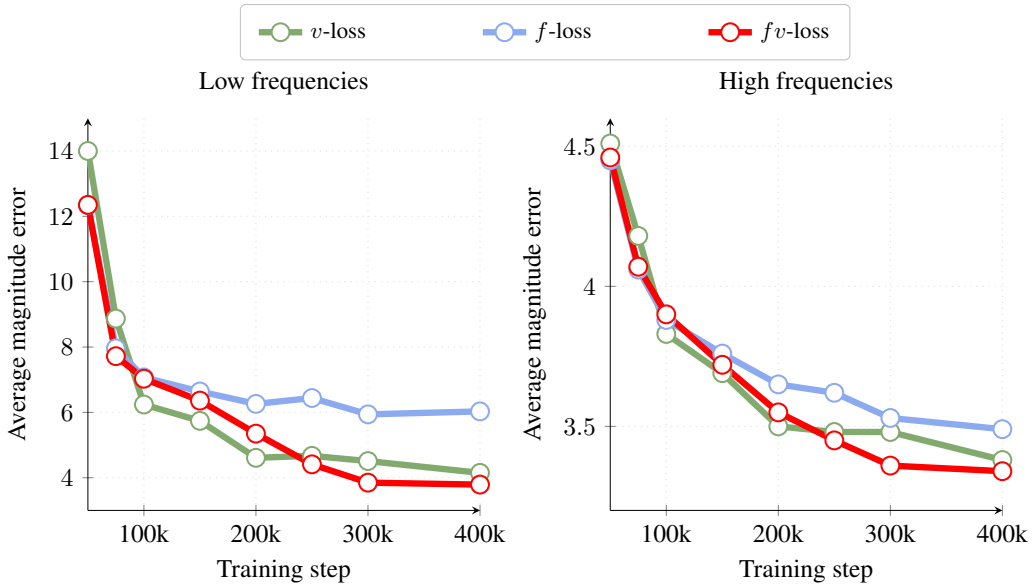


Figure 7: **Spectral error during training.** Average magnitude error in the low- (left) and high-frequency (right) bands. $\boldsymbol{f}$-loss corrects the spectral bias fastest early on but plateaus, while $\boldsymbol{fv}$-loss reaches the lowest error in both bands by the end of training.

C Toy Example

In section 3.2, we isolate the temporal dynamics of the spectral bias. We train a small MLP to generate synthetic images composed of exactly two frequency components: one low and one high. In addition to the evolution of the magnitude of the Fourier spectrum in Figure 2, we provide the raw spectrum in Figure 8.

D Focal Frequency Loss as a Continuous Laplacian Pyramid Loss

The Laplacian Pyramid [2] decomposes an image into bandpass residuals $\{b_0, ..., b_{K-1}\}$ and a low-pass base l_K. Each level captures spatial frequencies within a specific octave, treating scale-specific features as independent signals [15]. By performing recursive Gaussian blurring and subtraction, it isolates structural residuals at specific resolution scales, effectively partitioning the frequency spectrum into octave bands. By treating each octave (2^n to 2^{n+1}) as an independent signal, the pyramid elevates sparse, high-frequency details to the same structural importance as the low-frequency base — preventing high-energy components from numerically overwhelming fine textural details.

Generalizing Laplacian Pyramids We propose $\boldsymbol{f}$-loss as a continuous, infinite-band generalization of octave-based decomposition. By mapping images into the frequency domain via the DFT, we gain

Loss	SiT output	FID @ Epoch 80
v-loss	x-pred	41.35
f-loss	x-pred	**36.38**

Table 8: **Generalization to the latent-space SiT [39] backbone.** FID comparison between v-loss and f-loss, both using x-prediction. Neither reaches the FID of the original SiT model trained with v-prediction (FID 33 at epoch 80).

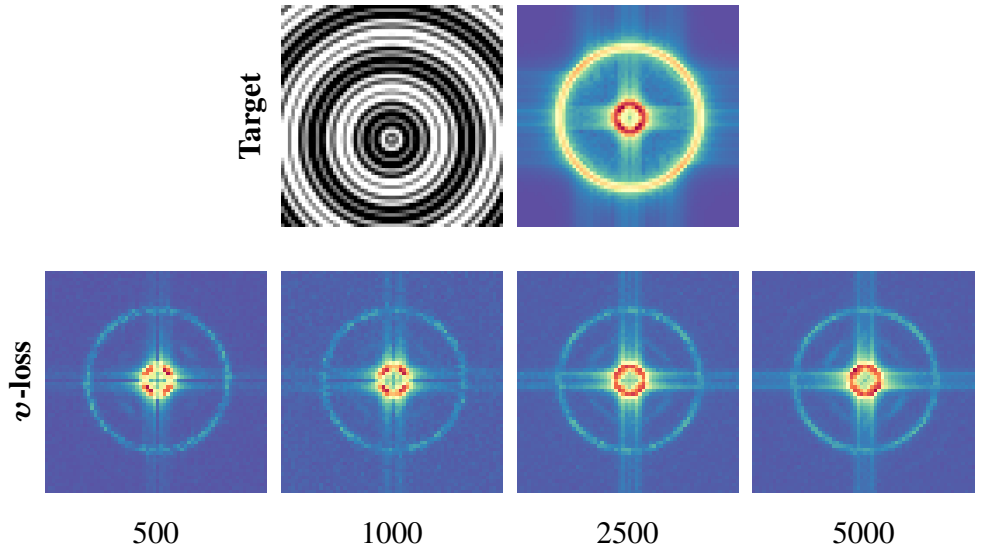


Figure 8: **Spectrum of the toy experiment of Section 3.** (top) Real image and its corresponding spectrum. (bottom) Spectrum of images generated with the MLP at various training steps. The low frequencies (inner ring) are learned much faster than the high frequencies (outer ring).

access to the full spectral hierarchy in a single pass. For a frequency at coordinate (u, v), we define:

$$\mathcal{L}_{u,v} = \log\left(1 + \|F_\theta(u,v) - F(u,v)\|_2\right) \tag{5}$$

The spectral difference $e_{u,v} = \|F_\theta(u,v) - F(u,v)\|_2$ serves as the functional equivalent of a Laplacian residual. The logarithmic mapping linearizes the hierarchical scale of the frequency spectrum: every doubling of frequency is granted equal structural weight, preventing the loss from being dominated by high-energy, low-frequency components. Our approach thus inherits the perceptual benefits of the Laplacian pyramid without the burden of explicit hierarchical branching.

Empirical study Table 9 reports results when transitioning from a discrete spectral decomposition to our continuous f-loss. We partition the Fourier domain into a fixed number of frequency bands

Loss	#Bands (Freq x Orient)	FID ↓ cfg=1.0	FID ↓ cfg=3.5	Prec. ↑	Rec. ↑	Den. ↑	Cov. ↑
v-loss	–	49.63	7.70	0.52	0.64	0.53	0.79
f-loss	2×1	45.93	8.78	0.53	0.64	0.55	0.81
	2×4	47.48	7.87	0.52	0.64	0.55	0.82
	3×4	44.81	6.89	0.54	0.65	0.57	0.82
	4×4	43.79	6.63	0.55	0.65	0.59	0.83
	∞	**41.78**	**6.39**	**0.56**	**0.66**	**0.62**	**0.85**

Table 9: Comparison between discrete & continuous Focal Frequency loss. (Note: The Precision, Recall, Density and Coverage are provided for unguided settings)

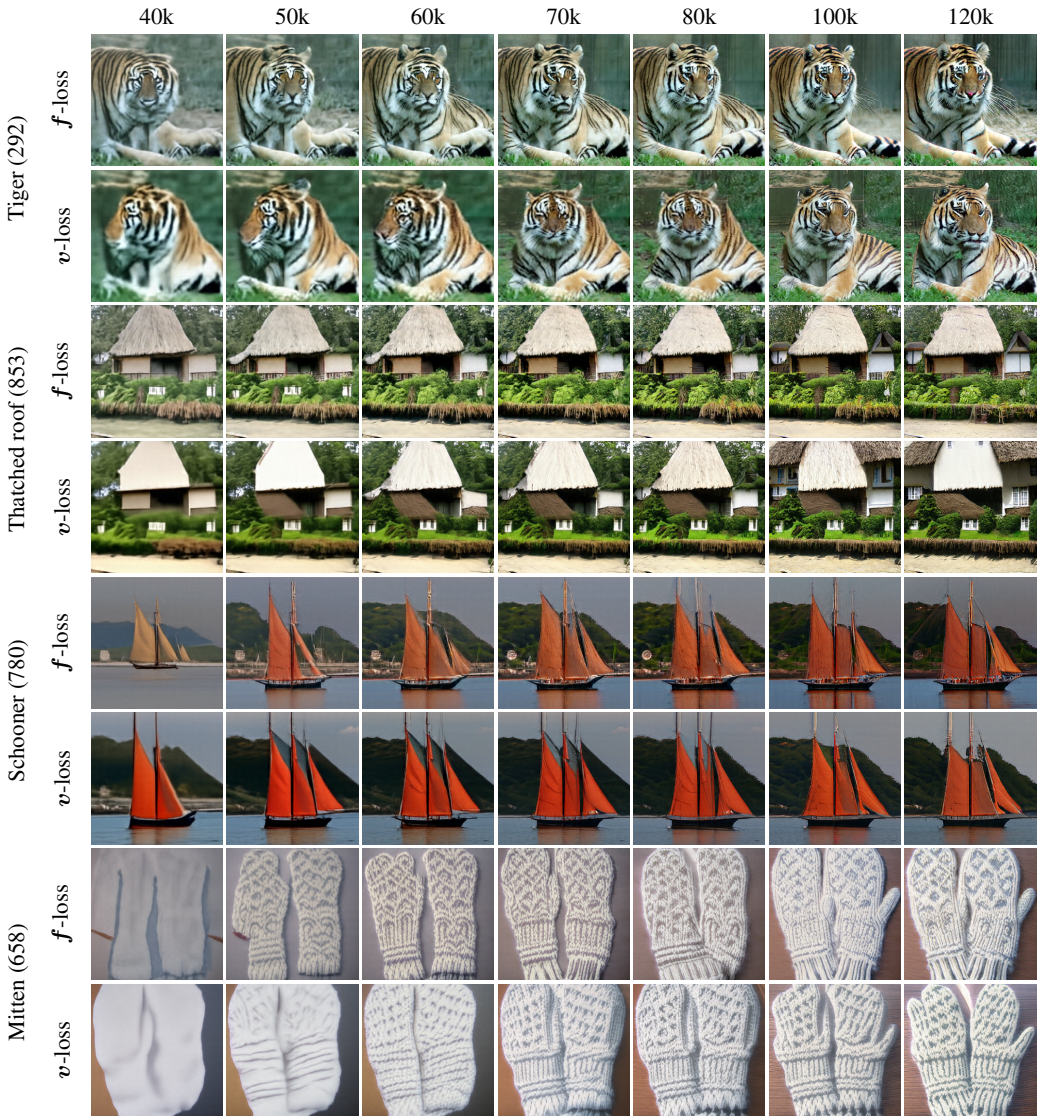


Figure 9: Extended comparison between $\boldsymbol{f}$-loss and $\boldsymbol{v}$-loss in the early stages of training.

($m = 2, 3, 4$) and orientations ($k = 1, 4$), computing the focal loss independently within each cell. Even a coarse discretization already improves over `v-loss`, and performance consistently improves as the number of spectral cells increases — from FID 47.48 at 2×4 bands to 43.79 at 4×4 bands. Our continuous formulation achieves the best performance with FID 41.78, confirming that fine-grained spectral weighting provides the most effective way to balance the learning signal across the image spectrum.

E Extended comparison in early stages of training

Figure 9 shows an extended quantitative comparison between $\boldsymbol{f}$-loss and $\boldsymbol{v}$-loss in the early stages of training, extanding Figure 5

F Additional qualitative results at 512×512 resolution

Figure 10 shows 512^2 images generated by our model trained with $\boldsymbol{fv}$-loss.



Figure 10: Additional Qualitative Results at 512^2 resolution. Model trained with $\boldsymbol{fv}$-loss for 400k steps.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction of training a better pixel flow matching better is supported via empirical results shown in Section 3 and Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The scope of the paper is empirical and, as such, doesn't involve any new theoretical contributions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method uses open-source Imagenet dataset and provides every details of the method in Section 3 and Section 4. The method builds upon the open-source `JiT` framework and in the abstract, clearly states that model and code will be open-sourced once the paper is accepted.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper explicitly states that the codes and model artifacts will be released after the paper's acceptance in the abstract. The paper uses the open-source Imagenet dataset in all the experiments, further focusing on the open-science scope of the research community.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides clear experimental settings in Section 4. Additionally in cases of when the paper motivates some engineering decision from prior works, it clearly cites the prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper provides benchmark scores on `FID`, `IS` and `FDD`. However, providing statistical error bars is computationally expensive for such metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper doesn't provide compute resources explicitly in the paper but given it builds upon prior works like `JiT` it implicitly uses similar compute budget as prior works.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conform to the NeurIPS code of ethics and uses open-source open-model frameworks. The paper clearly conform to double-blind anonymity for proper verification in the further reviewing process.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper promotes better and faster pixel flow matching model; directly pointing towards more efficient and resource frugal way to train such models. Though explicitly not mentioned, the paper clearly indicates a positive societal impacts by providing new way to train faster.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper is mainly focused on training scheme based on open-source dataset and open-source frameworks. There is always a risk misuse for generative models, but the method mainly focuses on new training paradigm and doesnt focus much on web-scaled data sources which might be vulnerable to much stronger safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper clearly cites all original owners wherever necessary, like the Imagenet dataset [50], JiT framework [34], etc.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper clearly states in the abstract that the newly trained models along with the code will be open-sourced.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper provides evaluation through automatic benchmarks like `FID`, `FDD`, etc and don't include any human evaluation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper doesn't involve any human study.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The paper used LLM only for writing, editing and formatting purposes and also for small standard implementations.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.