FastDINOv2: Frequency Based Curriculum Learning Improves Robustness and Training Speed

Jiaqi Zhang[1] **Juntuo Wang**[1] **Zhixin Sun**[2] **John Zou**[1] **Randall Balestriero**[1]

[1]Brown University [2]Cornell University
{jiaqi_zhang6, juntuo_wang, john_zou, randall_balestriero}@brown.edu
zs454@cornell.edu

Abstract

Large-scale vision foundation models such as DINOv2 boast impressive performances by leveraging massive architectures and training datasets. But numerous scenarios require practitioners to reproduce those pre-training solutions, such as on private data, new modalities, or simply for scientific questioning–which is currently extremely demanding computation-wise. We thus propose a novel pre-training strategy for DINOv2 that simultaneously accelerates convergence–and strengthens robustness to common corruptions as a by-product. Our approach involves a frequency filtering curriculum–low-frequency being seen first–and the Gaussian noise patching augmentation. Applied to a ViT-B/16 backbone trained on ImageNet-1K, while pre-training time and FLOPs are reduced by $1.6\times$ and $2.25\times$, our method still achieves matching robustness in corruption benchmarks (ImageNet-C) and maintains competitive linear probing performance compared with baseline. This dual benefit of efficiency and robustness makes large-scale self-supervised foundation modeling more attainable, while opening the door to novel exploration around data curriculum and augmentation as means to improve self-supervised learning models robustness. The code is available at `https://github.com/KevinZ0217/fast_dinov2`

Figure 1: The FastDINOv2 training pipeline comprises two stages. In the first stage, the initial 75% of training epochs utilize only low-frequency features extracted via downsampling. In the second stage, the remaining 25% of epochs employ full-resolution images with Gaussian noise patching.

Table 1: Comparison of training costs, evaluation on ImageNet-C, and linear probing on the ImageNet-1K validation set using a frozen ViT-B DINOv2 backbone trained on ImageNet-1K.

Training Method	Training Time (days)(↓)	ImageNet-1K(↑)	ImageNet-C (↓)	GFLOPs (↓)
DINOv2	16.64 (NVIDIA L40S)	77.8%	56.5%	493.76
FastDINOv2	**10.32 (NVIDIA L40S)**	**76.2%**	**56.7%**	**219.92**

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Models such as DINOv2 [22] and CLIP [25], built on Vision Transformer (ViT) backbones [11], achieve remarkable performance, generalization, and even upstream robustness [39] [24]. These advances stem largely from self-supervised learning (SSL)—a paradigm in which models learn useful representations from unlabeled data by solving proxy tasks (e.g., contrastive matching, predicting masked patches) rather than relying on costly manual annotation [12]. SSL's appeal lies in its ability to leverage the massive, ever-growing pools of raw images available online; by discovering structure in data itself, SSL yields features that transfer effectively to downstream tasks such as classification, objection detection, and segmentation, often matching or exceeding their fully supervised counterparts [9] [13].

However, reproducing these recent breakthroughs typically demands huge compute: large ViT variants (e.g., ViT-G) trained for hundreds of epochs on billions of images [4], multi-GPU clusters, and sophisticated optimization recipes. This resource barrier can put state-of-the-art SSL out of reach for many academic labs and startups, limiting both reproducibility and further innovation. Moreover, while pre-trained SSL models frequently demonstrate excellent performance on clean benchmarks, robustness to real-world distribution shifts—common corruptions, sensor noise, weather effects—remains crucial for safety-critical applications such as medical imaging [2] [30]or autonomous driving [4]. Although some studies have shown that SSL can improve robustness compared to supervised pre-training [18] [3] [21], most SSL methods do not explicitly optimize for robustness, and existing robust pre-training often compounds the compute demands [26] [35].

In particular, recent large-scale SSL models exhibit an emergent robustness only when trained at extreme scales: for instance, DINOv2 and related methods require model sizes upward of 86 millions of parameters for ViT-B and datasets of similar magnitude, such as LVD-142M and LAION-5B [28], to yield strong resistance to corruptions. This scale-driven robustness is appealing in principle, but is prohibitively expensive in practice for researchers without access to massive compute. Consequently, there is a pressing need to design compute-efficient SSL training method that still deliver robustness guarantees.

To address this gap, we propose a two-stage curriculum for DINOv2 pre-training that both speeds up convergence and enhances robustness to frequency-based corruptions—all without resorting to prohibitively large models or datasets. Our approach is motivated by the observation that high-frequency and low-frequency corruptions pollute different spectral bands of images, and that data-driven curricula can steer learning to emphasize or de-emphasize particular frequency bands at different training phases. The training curriculum consists of two stages:

Stage 1 – Low-frequency training. We begin by downsampling images, emphasizing their low-frequency components. This encourages the model to quickly learn broad, coarse features and accelerates convergence on clean data.

Stage 2 – Full-resolution with high-frequency augmentation. We then transition to full-resolution inputs while introducing Gaussian-noise patching, where random image patches are replaced with noise. This forces the model to learn invariances to high-frequency perturbations and improves robustness.

With extensive experiments, our method achieves not only faster convergence, but also higher robustness on models and datasets with various scales. This work demonstrates that robustness need not be an emergent by-product of extreme scale but can instead be built into SSL through careful curriculum design and augmentation. We believe this opens the door to more accessible, reproducible, and robust self-supervised training. In summary, our contributions are the following:

- We conduct a comprehensive robustness and frequency-based analysis on models pretrained with a low-frequency data curriculum—an underexplored direction. We identify the low-frequency bias introduced by this training scheme and propose Gaussian noise patching as a complementary augmentation to enhance robustness.
- The proposed curriculum accelerates convergence on ImageNet-1K [27] with DINOv2 and a ViT-B backbone, reducing pretraining time by $1.66\times$ and FLOPs by $2.25\times$, while maintaining matching robustness and competitive clean linear probing accuracy.

2 Related Work

Frequency-Guided Curriculum Learning By ordering the example from the easiest to the hardest, curriculum learning yields a monotonic increase in the rate of convergence under stochastic gradient descent [34]. It has also been proven that an ideal curriculum can effectively smooth the optimization landscape without changing the location of global minima [14]. Thus, defining the level of difficulty effectively is essential for a powerful curriculum. Gradient-based training of deep networks systematically fits the low-frequency components of a target function first, and gradually capture high-frequency features as training proceeds [36]. For vision transformer, specifically, the multi-head self-attention acts as a low-pass filter, attenuating high-frequency signals while preserving low-frequency components [23].

The coarse-to-fine paradigm in computer vision motivates training models first in images with reduced information, then on their full-resolution counterparts. Therefore, it is natural to define low-frequency components from an image as easy examples, while full image with both low- and high-frequency features as more difficult ones. From a Fourier perspective, natural image content is concentrated in the low-frequency domain [37], so downsampling preserves most of the information while reducing computational cost. Several works incorporate small images or low-frequency components into the training pipeline to accelerate ViT convergence: RECLIP[19] applies this curriculum to pre-train CLIP, and EfficientTrain++ [33] generalizes it for various ViT-based models, including MoCo [15] and MAE [16]. However, the impact of this approach on model robustness remains unclear. Moreover, as a general method, EfficientTrain++ evaluated it on DINO [7], but did not observe convergence speedup. In this paper, we revisit the curriculum learning applied to DINOv2. Besides the acceleration in training convergence, we also discover that this data curriculum unexpectedly biases model toward high-frequency information.

Frequency Bias to Robustness-Accuracy Tradeoffs Robustness research often adopts a frequency-domain perspective to examine how image corruptions correspond to the frequency spectrum [37]. In natural images, low-frequency components dominate, carrying most structural information, so high-frequency corruptions, such as Gaussian or shot noise, primarily disrupt fine details like edges, while low-frequency corruptions, including contrast shifts, brightness changes, or fog, modify broader patterns. Recent works have studied the need to model the noise as part of the data augmentation strategies–justifying the use of such corruptions as means to improve downstream robustness [31]. Different data augmentations introduce frequency biases that enhance robustness to specific corruptions but reveal distinct strengths and weaknesses [8]. For example, Gaussian noise augmentation strengthens robustness to high-frequency perturbations, producing a low-frequency-biased model, but often degrades performance on medium-frequency distortions like motion blur or Gaussian blur [37]. Similarly, CutOut [10] augmentation, which masks image regions to emphasize high-frequency cues such as edges or textures, encourages models to rely on features that are easily corrupted by noise or blur, limiting robustness gains. Adversarial training, on the other hand, biases models toward low-frequency features, enhancing resilience to certain perturbations but compromising performance on low-frequency corruptions [38] [6]. While these augmentations offer specific strengths, they often impair generalization by prioritizing certain frequency bands over others. In this work, we address this tradeoff by integrating Gaussian noise patching [20], which applies noise to localized image patches to balance robustness to high-frequency corruptions with preserved accuracy on clean images. This approach complements the high-frequency bias induced by our data curriculum, fostering a more balanced model across the frequency spectrum, though careful evaluation is needed to mitigate potential weaknesses on medium-frequency corruptions.

3 FastDINOv2: Recipe for Fast and Robust Pretraining

This section presents the proposed curriculum learning pipeline. At a high level, the pipeline incrementally introduces high-frequency features through a structured learning curriculum. Additionally, Gaussian noise patching is incorporated into the training process to enhance robustness.

3.1 Low-Frequency Feature Extraction

Low-frequency features can be extracted through multiple approaches. While methods like EfficientTrain++ [33] employ Fourier transform with high-frequency filtering for this purpose, we opt for

a simpler and more computationally efficient strategy. In our pipeline, we use downsampling as a lightweight proxy for low-frequency extraction, applied after DINOv2's standard image cropping preprocessing. In DINOv2's teacher-student framework, the teacher network processes global crops, which preserve low-frequency structural information, while the student network handles local crops, which retain high-frequency details. The cropping operation is defined as follows:

$$s \sim \mathcal{U}\big(s_{\min}, s_{\max}\big), \quad \tilde{w} = \tilde{h} = \sqrt{s\,H\,W} \tag{1}$$

where $s_{\min}(>= 0.08)$ and $s_{\max}(<= 1.0)$ define the crop's area ratio relative to the original image dimensions (H, W). We adopt DINOv2's original scaling parameters: global crops use $(s_{\min}, s_{\max}) = (0.32, 1.0)$, while local crops use $(0.05, 0.32)$.

After cropping, downsampling is performed to extract low-frequency features. This operation uses bicubic interpolation, a resampling method that computes a weighted average of the nearest 4×4 neighborhood of input pixels through a cubic convolution kernel. The specific definition for cubic kernel function is in A.3. We reduce global crop sizes from 224×224 to 112×112 and local crops from 96×96 to 48×48.

3.2 Frequency Based Curriculum Learning

Training Curriculum Curriculum learning [5] improves model performance by progressively training on simpler samples before advancing to more complex ones. In our approach, we define low-frequency components of images—representing coarse, large-scale patterns—as the easier samples, gradually progressing to harder examples represented by the original images. Our curriculum consists of two stages. For the first 75% of training epochs, the model is trained exclusively on these low-frequency components. Subsequently, we apply a restarting mechanism by resetting the Adam optimizer's training dynamics. In the second stage, we train DINOv2 on original images containing both low- and high-frequency information for the remaining epochs. To ensure training stability, we maintain a fixed batch size across both stages.

Balancing Frequency Bias via Gaussian Noise Patching As previously mentioned, training with the low-frequency curriculum biases the model toward high-frequency features, enhancing robustness against low-frequency corruption compared to the standard DINOv2 training approach. To balance this bias and improve robustness to low-frequency features, we introduce Gaussian noise patching in the second stage of training. Combined with cut-out augmentation and Gaussian noise augmentation, this technique effectively encourages the model to focus on low-frequency features while maintaining performance on clean data. For implementation, we randomly select a square patch for each image, with side length $\tilde{h} = \tilde{w} < \min(H, W)$, where H and W are the image's height and width before transformation. We then apply Gaussian noise to this patch, perturbing each pixel value x with a noise value $\tilde{x}$ independently sampled from a normal distribution $\tilde{x} \sim \mathcal{N}(1, \text{scale}^2)$. The noise intensity is controlled by the parameter scale.

4 Experiments

4.1 Datasets and Training Setup

Datasets For faster experimentation, we primarily use ImageNet-100 [32], a subset of ImageNet-1K [27]. The training set consists of 100 randomly selected classes from ImageNet-1K, with the first 500 images from each class. Similarly, the validation set contains the corresponding 100 classes from the original validation set, with 50 images per class. This results in a total of 50,000 training images and 5,000 validation images. For robustness evaluation, we use ImageNet-100-C, derived from ImageNet-C [17], which benchmarks model resilience to common corruptions. We maintain the exact image selection from the ImageNet-100 validation set across all corruption levels and types. Additionally, we employ ADE20K for semantic segmentation tasks. Finally, we scale our approach to full ImageNet-1K and evaluate robustness on ImageNet-C.

Model and Training Details For ImageNet-100 experiments, we use ViT-S/16 as the DINOv2 backbone with a total batch size of 40, distributed across 4 GPUs (10 per GPU). Experiments for large batch size in first stage of training are in Appendix A.1. We adopt positional embedding with

interpolation for adapting to image resolution shift between two training stages. The results of applying positional embedding with or without interpolation across different embedding sizes are in Appendix A.2. We train baseline models for 500 epochs and training curriculum experiments for 200 epochs, ensuring the baseline converges to optimal performance. All ImageNet-100 experiments use a fixed epoch length of 1,250 iterations. For ImageNet-1K experiments, we employ ViT-B/16 with a total batch size of 512 (128 per GPU), with epoch length of 2,500 iterations. The baseline and FastDINOv2 are trained for 250 and 200 epochs on ImageNet-1K, respectively. Following the official DINOv2 implementation, we use AdamW optimizer with square root learning rate scaling based on batch size, yielding a base learning rate of 7.9×10^{-4}. For linear probing evaluation, we use a batch size of 128 with 12.5k total iterations. All training runs are distributed across 4 NVIDIA L40S GPUs, while evaluations use either NVIDIA A6000 or NVIDIA A5500 GPUs.

4.2 Linear Probing Reveals Fast Convergence Without Compromising Accuracy

We evaluate our low-frequency data curriculum against the DINOv2 baseline through linear probing, training linear classifiers on frozen backbones. The baseline ViT-S/16 model follows standard DINOv2 training for 500 epochs with consistent 224×224 resolution. Our curriculum employs 112×112 images for the first 75% of training (150 epochs), then transitions to 224×224 images for the remaining epochs (denoted as 112-224 curriculum).

Notably, the 112-224 curriculum achieves the baseline's 250-epoch accuracy by epoch 200, demonstrating 20% faster convergence in training epochs and $1.44\times$ acceleration in training time. We additionally validate the framework's flexibility by testing alternative first-stage resolutions in Table 2, observing consistent efficiency gains across configurations.

The 112×112 resolution in our curriculum's initial phase reduces input tokens per forward pass by 75% compared to baseline (Table 2), yielding substantial computational savings. However, we identify a critical tradeoff: excessively small first-stage resolutions (e.g., 64×64 or 96×96) degrade linear probing performance due to insufficient learning signal. While these ultra-low resolutions marginally improve training speed, they exhibit diminishing returns in efficiency gains compared to 112×112. Thus, we select 112×112 as the optimal first-stage resolution - balancing computational efficiency with preserved information.

Table 2: Linear probing accuracy on ImageNet-100. Our data curriculum pipeline matches DINOv2 performance while achieving a $1.44\times$ convergence speed-up. 112-224 means that the FastDINOv2 training utilizes the 112-224 data curriculum; same applies to the remaining training methods. The training time is measured in NVIDIA L40S hours on a single GPU.

Training method	Accuracy	Training epoch	Training time
DINOv2	78.6%	250	24h
112-224 FastDINOv2 w/o GP	**78.44%**	200	13.9h
128-224 FastDINOv2 w/o GP	77.74%	200	13.6h
96-224 FastDINOv2 w/o GP	77.2%	200	12.9h
64-224 FastDINOv2 w/o GP	70.6%	200	13.48h

4.3 Data Curriculum Induces High-Frequency Feature Bias

While previous sections analyzed the curriculum design and its training acceleration benefits, we now investigate how low-frequency data curriculum impacts model robustness against common corruptions. Using ImageNet-C as our corruption robustness benchmark, we construct ImageNet-100-C by preserving corresponding classes and images from the ImageNet-100 validation set.

Our evaluation protocol trains linear classifiers exclusively on clean ImageNet-100 validation data using frozen backbones, then tests them on the corrupted ImageNet-100-C images. This ensures neither the model nor classifier encounters corrupted data during training. We compute average error rates across all corruption types and severity levels to quantify robustness.

In Table 3, our curriculum-trained models exhibit greater robustness to low-frequency corruptions compared to baseline, indicating stronger reliance on high-frequency features for classification. This emerges because high-frequency exposure during later training epochs forces the model to develop invariant representations of these components, thereby amplifying sensitivity to high-frequency

corruptions. The results confirm that our curriculum induces a high-frequency bias – an interesting product of the training strategy.

To better interpret this shift in robustness, we group corruption types in Table 3 based on their dominant frequency characteristics, following a Fourier analysis approach [37]:

- **Low-frequency corruptions:** Brightness, contrast, fog, frost
- **Mid-frequency perturbations:** Motion blur, defocus blur, glass blur, Gaussian blur, snow, zoom blur
- **High-frequency distortions:** Gaussian noise, impulse noise, shot noise, speckle noise
- **Hybrid effects:** Elastic transform, JPEG compression, pixelate, saturate, spatter

Table 3: ImageNet-100-C testing accuracies for DINOv2 baseline trained with 250 epochs, 112–224 data curriculum trained with 200 epochs, DINOv2 baseline trained with 200 epochs, and DINOv2 with Gaussian noise patching trained by 200 epochs. Comparison between DINOv2(250Ep) and 112-224(200Ep) suggests a low-frequency bias induced by data curriculum, while the comparison between DINOv2(200Ep) and DINOv2 w/ GP reveals the high-frequency bias from Gaussian noise patching.

	Test 112–224 Curriculum		Test Gaussian Patching(200 Ep)	
Corruption type	DINOv2(250Ep)	112-224 (200Ep)	DINOv2	DINOv2 w/ GP
Brightness	71.23%	**71.60%**	64.76%	**66.43%**
Contrast	51.49%	**55.24%**	45.83%	**47.75%**
Fog	47.06%	**48.22%**	38.42%	**40.18%**
Frost	43.80%	**45.45%**	36.58%	**39.95%**
Defocus blur	**42.01%**	40.24%	**33.80%**	32.69%
Gaussian blur	**44.38%**	41.94%	**37.15%**	35.36%
Glass blur	36.85%	**36.94%**	29.36%	**30.56%**
Motion blur	43.64%	**45.88%**	37.37%	**38.21%**
Snow	37.25%	**38.08%**	**30.22%**	28.84%
Zoom blur	47.26%	**47.40%**	**42.12%**	41.54%
Gaussian noise	**32.11%**	29.59%	21.72%	**48.72%**
Impulse noise	**26.97%**	25.62%	17.34%	**44.89%**
Shot noise	**31.06%**	29.04%	21.30%	**45.68%**
Speckle noise	**40.87%**	39.30%	31.25%	**50.46%**
Elastic transform	**62.92%**	62.07%	56.70%	**57.17%**
JPEG compression	**58.58%**	58.02%	51.71%	**52.44%**
Pixelate	**51.79%**	50.46%	**44.98%**	44.47%
Saturate	**65.59%**	64.89%	57.39%	**58.74%**
Spatter	55.12%	**55.34%**	48.58%	**49.60%**
Corruption accuracy	**46.84%**	46.60%	39.29%	**44.93%**
Clean accuracy	**78.60%**	78.42%	73.46%	**74.05%**

4.4 Spectral Balance: Combining Curriculum and Noise Patching

Building on the high-frequency feature bias observed in the training curriculum, we demonstrate how Gaussian noise patching, a low-frequency-biased augmentation, can counterbalance this effect. Importantly, integrating these approaches retains their individual strengths while addressing their respective limitations.

We identify Gaussian noise augmentation [1] as a method to introduce low-frequency bias, demonstrated by enhanced robustness against high-frequency noise corruptions. However, its direct application decreases clean accuracy due to excessive low-frequency emphasis. Gaussian noise patching mitigates this by selectively applying localized noise injection alongside preserved clean regions, thereby maintaining discriminative feature learning.

Table 3 compares the DINOv2 baseline with DINOv2 trained using Gaussian noise patching. The patched model exhibits a low-frequency bias, demonstrating greater robustness against high-frequency corruptions, including all noise corruptions. This frequency bias directly opposes that of our data curriculum, prompting the question: can Gaussian noise patching be integrated to balance the curriculum's frequency bias?

When applied solely during the high-resolution phase of the curriculum, this hybrid approach achieves balanced robustness improvements. As shown in Table 4, most corruption types show enhanced resilience, with minor reductions confined to specific mid-frequency distortions (e.g., zoom blur, elastic transform) and high-frequency artifacts (e.g., pixelate). Notably, the combined method eliminates conflicting vulnerabilities observed in individual applications, significantly improving robustness to defocus blur and Gaussian blur compared to either technique alone.

The synergy stems from complementary frequency interactions: the curriculum's early low-resolution training builds robust low-frequency representations, while subsequent Gaussian noise patching mitigates overfitting to artificial high-frequency patterns. This integration achieves a spectral balance, ensuring neither frequency domain overly dominates feature encoding.

These findings underscore the value of aligning augmentation strategies with curriculum stages. Although trade-offs remain for certain corruption types, this framework offers a pathway to balance spectral biases while maintaining core model performance.

Table 4: ImageNet-100-C test accuracy for DINOv2 baseline versus DINOv2 with data curriculum and Gaussian noise patching. Accuracy improves for most corruption types, demonstrating the effectiveness of combining data curriculum and Gaussian noise patching.

Corruption type	DINOv2	FastDINOv2	Δ
Brightness	71.23%	**71.92%**	+0.69%
Contrast	51.49%	**56.09%**	+4.60%
Fog	47.06%	**47.90%**	+0.84%
Frost	43.80%	**47.52%**	+3.72%
Defocus blur	42.01%	**42.31%**	+0.30%
Gaussian blur	44.38%	**44.44%**	+0.06%
Glass blur	36.85%	**40.24%**	+3.39%
Motion blur	43.64%	**45.43%**	+1.79%
Snow	37.25%	**37.96%**	+0.71%
Zoom blur	**47.26%**	47.00%	–0.26%
Gaussian noise	32.11%	**57.51%**	+25.40%
Impulse noise	26.97%	**54.62%**	+27.65%
Shot noise	31.06%	**55.34%**	+24.28%
Speckle noise	40.87%	**61.59%**	+20.72%
Elastic transform	**62.92%**	62.36%	–0.56%
JPEG compression	58.58%	**61.15%**	+2.57%
Pixelate	**51.79%**	49.18%	–2.61%
Saturate	65.59%	**65.74%**	+0.15%
Spatter	55.12%	**56.35%**	+1.23%
Corruption accuracy	46.84%	**52.88%**	+6.04%
Clean accuracy	**78.6%**	78.4%	–0.20%

4.5 Instance Recognition Indicates Instance-Level Effectiveness

Building on the demonstrated performance in linear probing, we further evaluate FastDINOv2 on an instance-level task through instance recognition. Specifically, we assess three difficulty levels on the Oxford and Paris datasets and report the mean average precision (mAP) in table 5. On the Oxford dataset, FastDINOv2 outperforms DINOv2 by 3.73% on the easy split and 1.22% on the medium split, with a slight drop on the hard split. Both models exhibit comparable performance on the Paris dataset. These results indicate that FastDINOv2 not only accelerates convergence but also enhances performance across a variety of tasks.

Table 5: Mean Average Precision (mAP) on instance-level recognition tasks using Oxford and Paris datasets (E: Easy, M: Medium, H: Hard)

Training method	Oxford			Paris		
	E	M	H	E	M	H
FastDINOv2	32.11%	22.26%	3.99%	56.38%	40.71%	14.56%
DINOv2	28.38%	21.04%	4.56%	56.88%	40.76%	14.94%

4.6 Semantic Segmentation Suggests Intact Pixel-Level Performance

Section 4.2 and 4.5 show that the FastDINOv2 achieves faster convergence while maintaining linear probing and improving instance recognition accuracy, proving that image and instance-level task remains strong. However, since semantic segmentation requires fine-grained pixel understanding, we needed to verify whether our approach - which spends 75% of training time on low-frequency features - could still perform well on such tasks. To test this, we conducted semantic segmentation experiments using ADE20K. We froze the DINOv2 backbone (trained with our curriculum) and trained only a 2-layer convolutional decoder, evaluating performance using both mean Intersection-over-Union (mIoU) and mean class accuracy (mAcc) metrics. Table 6 demonstrates that our 112-224 curriculum with Gaussian patching matches the DINOv2 baseline's segmentation performance (mIoU/mAcc) in just 200 epochs. Despite initial low-frequency training, the second stage successfully recovers fine-grained pixel understanding needed for segmentation tasks.

Table 6: Semantic segmentation evaluation with frozen DINOv2 backbone. A 2-layer convolutional decoder is trained on top of frozen DINOv2 using ADE20K dataset for 200 epochs.

Training Method	Pre-training Epochs	Clean Accuracy	mAcc	mIoU
DINOv2	250	78.6%	25.09%	19.2
FastDINOv2	200	78.4%	25.21%	19.16

4.7 Frequency Based Analysis and Grad-CAM Visual Explanation

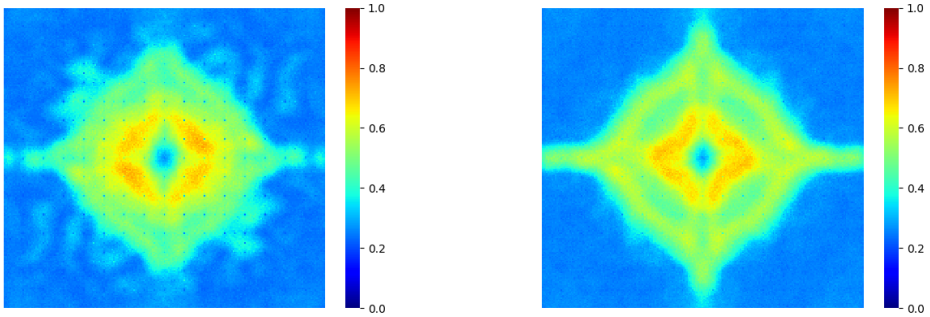
In addition to the evaluations using ImageNet-100-C and linear probing, we explore whether different training paradigms influence the error sensitivity of models across all frequency bands—not only those in ImageNet-100-C—and the regions the model prioritizes during predictions. To this end, we generate Grad-CAM visualizations [29] and Fourier spectral heatmaps following the methods in [37].

Figure 2 shows the error sensitivity of the resulting model across various frequency ranges. Compared to the DINOv2 baseline, our approach achieves lower error rates in high- and medium-frequency bands. In low-to-mid frequency ranges, particularly in the central region, our method reduces the error rate by a significant margin. However, the model shows increased susceptibility to perturbations in the mid-to-high frequency range, indicating potential trade-offs in robustness across frequency bands.

Figure 3 presents Grad-CAM visualizations that highlight further differences. While the DINOv2 baseline distributes its focus across both the target object and background areas during predictions, the model trained with FastDINOv2 emphasizes object contours more strongly. This suggests that the initial stage of pre-training provides an effective initialization, enabling the model to focus on coarse image components.

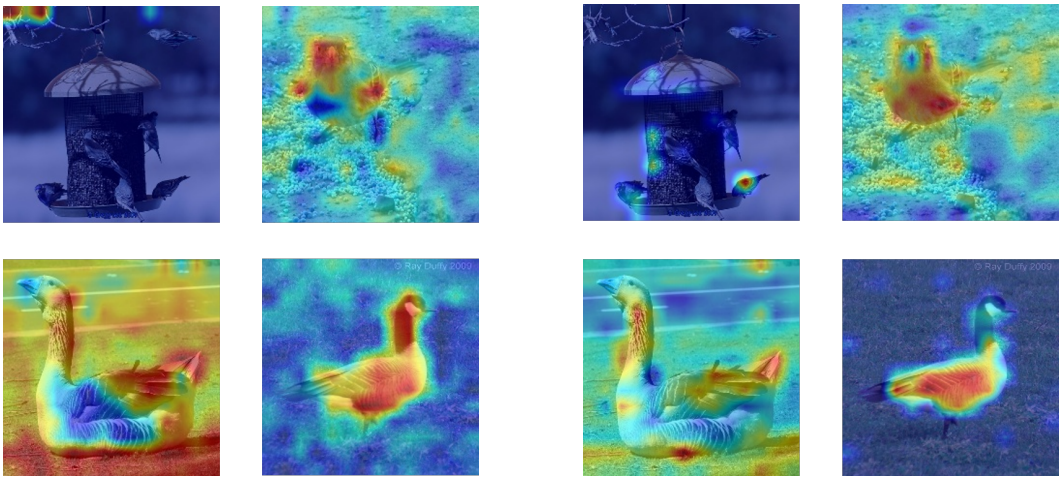
4.8 Extending to ViT-B and ImageNet-1K: Balancing Efficiency and Robustness

To validate scalability, we extend our framework to ViT-B architectures and ImageNet-1K pretraining. The baseline DINOv2 model is pretrained for 200 epochs using standard protocols, while our FastDINOv2 method combines a 112-224 curriculum with Gaussian noise patching over an equivalent number of epochs—allocating 150 epochs to downsampled inputs followed by 50 epochs of full-resolution training. Evaluation includes linear probing on clean ImageNet-1K validation data and robustness assessment using ImageNet-C with frozen backbones.



DINOv2 baseline FastDINOv2

Figure 2: Fourier error sensitivity heatmap of model trained with our method and DINOv2 baseline. The heatmap is generated with a subset of Imagenet-100 validation set with 5 images sampled from each class. Color indicates the error sensitivity to that specific frequency range. Low-frequency features are mainly concentrated into center area, while the region further away from center represents feature with high frequency.



DINOv2 baseline FastDINOv2

Figure 3: Grad-CAM maps for DINOv2 baseline and FastDINOv2. The first row images are from class "house finch, linnet, Carpodacus mexicanus", and the second row from "goose". With the data curriculum, model can better capture the contour of the object. See more examples in A.7

Table 1 reports the evaluation results and training costs for the baseline and our method. Although training time is reduced by 1.6× and FLOPs by 2.25×, our method achieves competitive linear probing accuracy. Moreover, testing on ImageNet-C with a frozen backbone and linear classifier shows a comparable error rate despite a slight drop in clean accuracy, demonstrating the effectiveness of our method in enhancing robustness.

Huge memory consumption for GPU is a critical bottleneck for scaling up the pretraining. However, with low-resolution image in the first training stage, FastDINOv2 allows a much lower memory requirement. As in table 7, during the first 75% of training epochs the maximum memory consumption of FastDINOv2 is 9.47GB for batch size of 128 per GPU, significantly less than 33.5GB for DINOv2 baseline with same batch size. This reduction in memory requirement shows possibilities for a cost-effective model by leveraging GPUs with lower memory capacity for most of the pretraining epochs.

Table 7: Comparison of training epochs and max memory consumptions on a single NVIDIA L40S GPU with 48GB memory. Both DINOv2 and FastDINOv2 apply ViT-B backbone trained on ImageNet-1K.

Training Method	Training Epochs		Max Memory Consumption(GB)	
	Low-Res	Full-Res	Low-Res	Full-Res
DINOv2	-	200	-	33.5
FastDINOv2	150	50	9.47	33.5

5 Limitations, Future Work, and Conclusion

In this work, we presented an efficient training framework for the vision foundation model DINOv2, demonstrating its scalability and robustness across diverse datasets and model backbones. Our method achieves substantial improvements in training efficiency, computational cost reduction, and model performance. However, one limitation of our current approach lies in the fixed schedule for transitioning from low-resolution to standard-resolution images during the final 25% of training epochs. This heuristic, while effective in our experiments, may not be optimal across different hyperparameter settings or training regimes. Future work will focus on developing adaptive scheduling strategies that dynamically determine the transition point based on training signals such as convergence metrics or validation performance. Additionally, we plan to investigate how our resolution-aware curriculum can be combined with other training enhancements such as adversarial training paradigms.

References

[1] M. Eren Akbiyik. Data augmentation in training cnns: Injecting noise to images. *arXiv preprint arXiv:2307.06855*, 2023. URL `https://arxiv.org/abs/2307.06855`.

[2] Mohammed Baharoon, Waseem Qureshi, Jiahong Ouyang, Yanwu Xu, Abdulrhman Aljouie, and Wei Peng. Evaluating general purpose vision foundation models for medical image analysis: An experimental study of dinov2 on radiology benchmarks. *arXiv preprint arXiv:2312.02366*, 2023. URL `https://doi.org/10.48550/arXiv.2312.02366`.

[3] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari S. Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Grégoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. URL `https://arxiv.org/abs/2304.12210`.

[4] Merve Rabia Barın, Görkay Aydemir, and Fatma Güney. Robust bird's eye view segmentation by adapting dinov2. In *Proceedings of the 2nd Workshop on Vision-Centric Autonomous Driving at ECCV 2024*, 2024.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[6] Qingwen Bu, Dong Huang, and Heming Cui. Towards building more robust models with frequency bias. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. URL `https://arxiv.org/abs/2104.14294`.

[8] Alvin Chan, Yew-Soon Ong, and Clement Tan. How does frequency bias affect the robustness of neural image classifiers against common corruption and adversarial perturbations? In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2022.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[10] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. URL `https://arxiv.org/abs/1708.04552`.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 2022. doi: 10.1109/MSP.2021.3134634.

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[14] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[18] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, 2019.

[19] Runze Li, Dahun Kim, Bir Bhanu, and Weicheng Kuo. RECLIP: Resource-efficient CLIP by training with small images. *arXiv preprint arXiv:2304.06028*, 2023. URL `https://arxiv.org/abs/2304.06028`.

[20] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. URL `https://arxiv.org/abs/1906.02611`.

[21] Wenquan Lu, Jiaqi Zhang, Hugues Van Assel, and Randall Balestriero. Ditch the denoiser: Emergence of noise robustness in self-supervised learning from data curriculum. *arXiv preprint arXiv:2505.12191*, May 2025. URL `https://arxiv.org/abs/2505.12191`.

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[23] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[24] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, 2022.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[26] Colorado J. Reed, Xiangyu Yue, Ani Nrusimha, Sayna Ebrahimi, Vivek Vijaykumar, Richard Mao, Bo Li, Shanghang Zhang, Devin Guillory, Sean Metzger, Kurt Keutzer, and Trevor Darrell. Self-supervised pretraining improves self-supervised pretraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. URL https://arxiv.org/abs/2210.08402.

[29] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[30] Xinrui Song, Xuanang Xu, and Pingkun Yan. Dino-reg: General purpose image encoder for training-free multi-modal deformable medical image registration. In *Medical Image Computing and Computer Assisted Intervention*, 2024.

[31] Hugues Van Assel, Mark Ibrahim, Tommaso Biancalani, Aviv Regev, and Randall Balestriero. Joint-embedding vs reconstruction: Provable benefits of latent space prediction for self-supervised learning. *arXiv preprint arXiv:2505.12477*, May 2025. URL https://arxiv.org/abs/2505.12477.

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

[33] Yulin Wang, Yang Yue, Rui Lu, Yizeng Han, Shiji Song, and Gao Huang. Efficienttrain++: Generalized curriculum learning for efficient visual backbone training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[34] Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[35] Cong Xu, Dan Li, and Min Yang. Adversarial momentum-contrastive pre-training. *arXiv preprint arXiv:2012.13154*, 2020. URL https://arxiv.org/abs/2012.13154.

[36] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019. URL https://arxiv.org/abs/1901.06523.

[37] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, 2019.

[38] Kejia Zhang, Juanjuan Weng, Yuanzheng Cai, Zhiming Luo, and Shaozi Li. Mitigating low-frequency bias: Feature recalibration and frequency attention regularization for adversarial robustness. *arXiv preprint arXiv:2407.04016*, 2024. URL https://arxiv.org/abs/2407.04016.

[39] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Anima Anandkumar, Jiashi Feng, and Jose M. Alvarez. Understanding the robustness in vision transformers. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

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 main claims are made explicitly in both abstract and introduction: we propose a efficient training pipeline for accelerating pre-training and improving robustness for vision foundation model such as DINOv2. Results in experiments section reflect the fast convergence, as well as how each component of the pipeline impacts model robustness.

   Guidelines:

   - The answer NA 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 NA 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 limitation is discussed in section 5, where we reflect on the limitation of determining transition epochs for our curriculum during pretraining.

   Guidelines:

   - The answer NA 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: [NA]

Justification: Our work does not include a formal theoretical result. The contribution from this work is algorithmic and experimental.

Guidelines:

- The answer NA 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: This paper provides comprehensive details for reproducing the experiments. The dataset and models used for experiments include ViT-S/16, ViT-B/16, Imagenet-1K, Imagenet-C, and ADE20K. All models and datasets above are publicly available, and more training details are included in experiment and method sections.

Guidelines:

- The answer NA 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 model and training dataset are all accessible in public, and we will release training code in supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides comprehensive details on hyperparameters, optimizer, epochs, and other training configurations.

   Guidelines:

   - The answer NA 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: [Yes]

   Justification: Our work does not report error bars for large experiments due to its computationally intensive nature. However, the experiment results are highly unlikely to be from chance as the findings are robust and have been observed in multiple settings with different model architecture, parameter, dataset, etc.

Guidelines:

- The answer NA 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: [Yes]

   Justification: We have provided detailed specification on the computational resources needed to run the experiments.

   Guidelines:

   - The answer NA 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: We carefully follow the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The goal of our work is to improve model pretraining efficiency and robustness. While there might be many potential societal consequences, none of which we feel must be specifically discussed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not have such risks.

Guidelines:

- The answer NA 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: All the external datasets and pre-trained model are preperly cited int the paper. For example, we have cited DINOv2, Imagenet-1K and Imagenet-C in the abstract and introduction sections.

Guidelines:

- The answer NA 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: Documentation and code for reproducing the result of all the experiments are provided in supplementary material.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This paper does not include any crowdsourcing and research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

Justification: This paper does not include any corwdsourcing and research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core methods of this paper does not involve usage of LLMs as any important, originall, or non-standard components.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 Constant Batch Size Ensures Training Stability

Smaller image sizes consume less GPU memory during training, which naturally suggests using a larger batch size in the first stage of the curriculum. To investigate the effect of an early large batch size, we test the 112–224 curriculum by increasing the batch size from 40 to 60 in the first stage while keeping the second stage unchanged. The base learning rate for the first stage is also scaled proportionally from 4×10^{-3} to 6×10^{-3}. As shown in Table 8, while the training time is slightly reduced with the early large batch size and scaled learning rate, the clean accuracy decreases. This suggests that varying the batch size during training can introduce instability and degrade model performance. Therefore, we use a constant batch size in all subsequent experiments.

Table 8: Linear probing accuracy for the early large batch size and learning rate. Notice that in this experiment, FastDINOv2 training utilizes 112-224 data curriculum and does not apply the interpolation for positional embedding and Gaussian patching in order to exclude their effects. In the Batch size column, $60 + 40$ indicates that batch size of 60 is used in the first stage, and 40 for second stage. Same for learning rate. Training cost is measured in NVIDIA L40S hours for a single GPU.

Training method	Accuracy	Training cost(hours)	Batch size	Learning rate
DINOv2	78.6%	250 epochs, 24	40	$4e-3$
FastDINOv2 w/o GP	78.06%	200 epochs, 12.3	60 + 40	$6e-3$ + $4e-3$
FastDINOv2 w/o GP	78.42%	200 epochs, 13.9	40	$4e-3$

A.2 Positional Embedding for Varying Image Resolution

DINOv2 uses learnable positional embeddings, which capture the location information of input patches during training. In the first half of the training curriculum, images are downsampled to 112×112 as input, while in the second stage, regular-sized images (224×224) are used. This setup requires the model to adapt to multiple input resolutions. Two approaches are available: either using separate positional embeddings for each stage or using a consistent positional embedding and interpolating it. The second approach introduces two further choices: interpolating the positional embedding based on either the first stage's smaller resolution (112×112) or the second stage's full resolution (224×224). In Table 9, we compare these settings and find that interpolation based on the first stage's smaller resolution (112×112) performs best. After the first training stage, we retain the positional embedding and apply bicubic interpolation for upsampling, adjusting it to the higher resolution in the second stage. This process mirrors our image downsampling procedure, where we also use a bicubic kernel to resample pixels.

Table 9: Linear probing accuracy on Imagenet-100 for FastDINOv2 with 112-224 data curriculum to compare the following three design: interpolation with fixed 112×112 positional embedding, interpolation with fixed 224×224 positional embedding, and using separate positional embeddings(no reusing). In the third design option, positional embedding from the first stage is abandoned, while a new positional embedding is trained based on second-stage's image resolution. FastDINOv2 training in this experiment does not use Gaussian patching to eliminate its effect.

Training method	Accuracy	Training epoch	Positional embedding
DINOv2	78.6%	250	no interpolation
FastDINOv2 w/o GP	**78.8%**	200	interpolation w/ 112×112
FastDINOv2 w/o GP	77%	200	interpolation w/ 224×224
FastDINOv2 w/o GP	78.42%	200	no reusing

A.3 Down-sampling with Bicubic Interpolation

We adopt Bicubic interpolation for down-sampling to obtain low-frequency inputs for first-stage training. The cubic kernel $w(t)$ is defined as:

$$w(t) = \begin{cases} (a+2)\,|t|^3 \;-\; (a+3)\,|t|^2 \;+\; 1, & |t| \le 1, \\ a\,|t|^3 \;-\; 5a\,|t|^2 \;+\; 8a\,|t| \;-\; 4a, & 1 < |t| < 2, \quad (a = -0.5) \\ 0, & |t| \ge 2, \end{cases} \tag{2}$$

where t represents the normalized distance from the sampling point to a neighboring pixel, and a controls the curvature of the cubic kernel. Using this kernel function, the bicubic interpolation $f(x, y)$ is computed by applying the convolution kernel w separably along both spatial dimensions:

$$f(x,y) = \sum_{m=-1}^{2} \sum_{n=-1}^{2} w\big(m - \Delta x\big)\; w\big(n - \Delta y\big)\; f\big(x_0 + m,\; y_0 + n\big); \quad m, n \in \{-1, 0, 1, 2\} \tag{3}$$

For an input image of size $(H_{\text{orig}}, W_{\text{orig}})$ and target downsampled size $(H_{\text{down}}, W_{\text{down}})$, the coordinate mapping transforms each pixel location (u, v) in the original image to (x, y) in the downsampled space as:

$$(x, y) = (u\,\frac{W_{\text{orig}}}{W_{\text{down}}}, v\,\frac{H_{\text{orig}}}{H_{\text{down}}}), \quad (x_0, y_0) = (\lfloor x \rfloor, \lfloor y \rfloor), \quad (\Delta x, \Delta y) = (x - x_0, y - y_0) \tag{4}$$

A.4 Experiments on SimCLR

In order to examine the generalization ability of our method across frameworks and model architectures, we conducted additional experiments by training a SimCLR model with ResNet backbone using our FastDINOv2 method (denoted as "FastSimCLR"). Using the LARS optimizer, the SimCLR baseline was trained for 200 epochs on full-resolution images from ImageNet-100, while FastSimCLR was trained for 150 epochs on low-frequency data followed by 50 epochs on full-resolution images. Despite the differences in inductive biases between ViTs and CNNs, in table 10 FastSimCLR shows faster convergence. Further, combining this curriculum with Gaussian noise patching improves robustness toward Imagenet-C.

Table 10: Linear probing accuracy on ImageNet-100 and testing accuracy on ImageNet-100-C for SimCLR and FastSimCLR. Training time is measured in NVIDIA A6000 hours for a single GPU.

Training Method	Epochs	ImageNet-100 (↑)	ImageNet-100-C (↑)	Training Time
SimCLR	200	64.48%	30.95%	14
FastSimCLR	150+50	64.82%	38.73%	11.2

A.5 Ablation Experiment for Transition Epoch

One critical design for FastDINOv2 pre-training framework is the transitioning epochs, where training data shifts from low-resolution images to full-resolution ones. Correctly choosing this transition point is essential - if the training epoch for the first stage is insufficient, the model fails to take advantage from low-resolution images to improve training efficiency; on the other hand, the model lacks exposure to the full-resolution features necessary for downstream tasks if the first stage is excessive. Thus, we determine this transition point empirically by conducting a set of cross validation experiments with different transition epoch in table 11.

Among all the splits, the 75/25 splits consistently yielded the best trade-off, achieving the highest linear probing accuracy. More extreme cases such as 0/100 would have no aceleration effect, and for 100/0 the model would only be trained on low-frequency data, which lead to notably worse performance.

Table 11: Linear probing accuracy on ImageNet-100 for FastDINOv2 with different transition epochs proportion. The number of total training epoch is 200, and training time is measured in NVIDIA A6000 hours for a single GPU.

Training Method	Split (Stage1/Stage2)	Training Time	ImageNet-100
FastDINOv2	65/35	20.4	77.6%
FastDINOv2	75/25	20.4	78.1%
FastDINOv2	85/15	20.68	76.8%
FastDINOv2	0/100	22.3	73.9%
FastDINOv2	100/0	20.1	69.7%

A.6 Convergence Speed Comparison

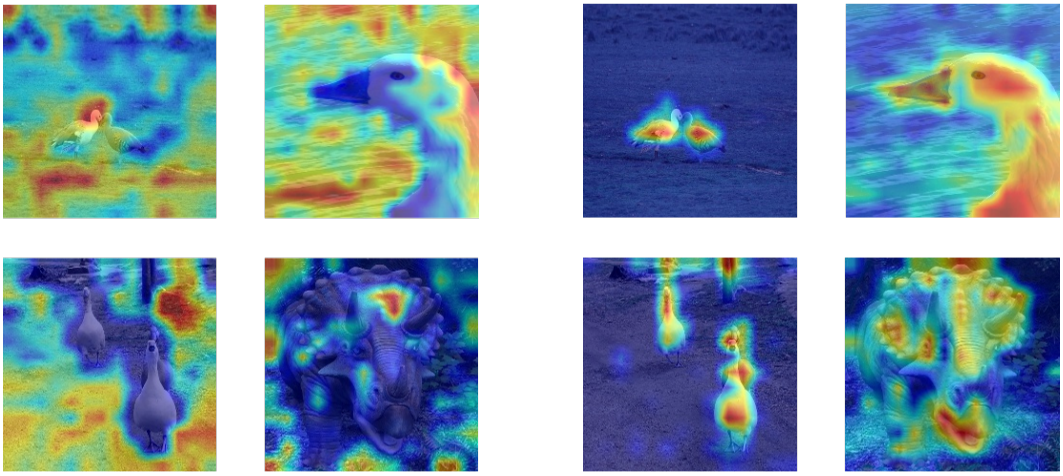
In order to better demonstrate the trend of convergence speed of FastDINOv2, we measure the training time gap for reaching several linear probing accuracy levels for both FastDINOv2 and DINOv2 baseline in table 12.

Table 12: Training time comparison (in NVIDIA A6000 hours) required to reach different accuracy levels on ImageNet-100.

Training Method	55% Accuracy	65% Accuracy	75% Accuracy	78% Accuracy
DINOv2	2.6h	4.8h	12.48h	23.04h
FastDINOv2	2.64h	5.28h	11.5h	13.1h

During the low-frequency training stage (when FastDINOv2 reaches 55% and 65% accuracy), FastDINOv2 achieves similar convergence speed to the baseline trained on full-resolution images. This is attributed to the inherent low-frequency bias in deep neural networks and transformers, such that the model is more inclined to learn low-frequency information first during training. Furthermore, this low-frequency stage provides a strong foundation for the subsequent full-resolution training stage (when FastDINOv2 reaches 75% and 78% accuracy), enabling FastDINOv2 to more efficiently and effectively learn fine-grained, high-resolution details from full-resolution images.

A.7 GradCAM Visualization and Imagenet-C Examples



DINOv2 baseline FastDINOv2

Figure 4: Extra Grad-CAM maps examples for DINOv2 baseline and FastDINOv2.

Figure 5: Imagenet-C examples for each corruption type.