Deep Learning Project Report

Yujia Zhang
New York University
yz6553@nyu.edu

Haojie Yin
New York University
hy2108@nyu.edu

Zihua Xin
New York University
zx1757@nyu.edu

Abstract

We study self-supervised pretraining of Vision Transformers (ViTs) under the computational and data constraints of the Fall 2025 Deep Learning Challenge. Using a fixed 96×96 *input resolution and fewer than 100M parameters, we compare two approaches for representation learning: a two-stage pipeline that combines masked autoencoding (MAE) with DINOv1 self-distillation, and a single-stage DINOv2 baseline. After pretraining, all encoders are frozen and evaluated on multiple downstream classification datasets using k-NN and linear probing. Experimental results show that the MAE→DINOv1 pipeline consistently outperforms the DINOv2 baseline across all test splits, highlighting the effectiveness of combining reconstruction-based pretraining with subsequent invariance-based refinement in moderate-scale SSL settings.*

1. Introduction

Our submission model checkpoints, training code, and training logs are saved at the following location.[1]

Self-supervised learning (SSL) has become a widely used paradigm for learning visual representations without labeled data. Recent methods such as masked autoencoders (MAE) and DINO-style self-distillation have shown strong performance, but their effectiveness under moderate-scale training regimes remains less explored. In particular, the Fall 2025 Deep Learning Challenge constrains models to a fixed input resolution of 96×96, fewer than 100M parameters, and a pretraining set of 500,000 unlabeled images, raising practical questions about which SSL objectives are most suitable in this setting.

Motivated by these constraints, we investigate two representative SSL strategies using Vision Transformer (ViT) backbones. The first is a two-stage pipeline that applies MAE pretraining followed by DINOv1 refinement, combining reconstruction-based learning with invariance-based self-distillation. The second is a single-stage DINOv2 baseline that relies exclusively on multi-crop self-distillation. All models share comparable architectural configurations to ensure a fair comparison.

After self-supervised pretraining, encoders are frozen and evaluated on standard downstream classification benchmarks using both k-NN and linear probing. This protocol isolates the effect of the SSL objective from downstream optimization and enables a direct comparison of representation quality under identical constraints.

2. Methodology

2.1. Overall Pipeline

We used a unified self-supervised learning and evaluation pipeline across all SSL methods attempted. One encoder is first pretrained from scratch on unlabeled images using a self-supervised learning objective. Depending on the method, this pretraining consists of either a single-stage self-distillation approach or a two-stage procedure combining masked image modeling and self-distillation. After self-supervised pretraining, the encoder is frozen and used as a feature extractor for downstream evaluation via k-NN classification and linear probing.

2.2. Self-Supervised Learning Methods

2.2.1. MAE + DINOv1 (Our Submission Model)

This method employs a two-stage self-supervised learning strategy that combines masked autoencoding with DINO-based self-distillation. Both stages share the same Vision Transformer (ViT) encoder architecture to ensure compatibility between reconstruction-based pretraining and subsequent representation refinement.

Motivation and Intuition. Different self-supervised learning objectives emphasize complementary aspects of visual representations. Masked autoencoding (MAE) focuses on reconstructing missing image content, encouraging the encoder to capture low-level structure and spatial context. In contrast, DINO-style self-distillation promotes semantic invariance by aligning representations of

[1] https://drive.google.com/drive/folders/18DPgaaydkoY7bN9345XOS5pRv3VBd0EM?usp=sharing

multiple augmented views through a student–teacher framework. By combining these two objectives sequentially, the encoder is first guided to learn structural image features and is then refined to produce semantically meaningful and augmentation-invariant representations.

Two-Stage Training Procedure. The self-supervised pretraining process consists of the following stages:

- **Stage 1: MAE Pretraining.** In the first stage, the encoder is pretrained using masked autoencoding (MAE), where a subset of image patches is masked and the model is trained to reconstruct the missing content from visible patches.
  - We adopt the MAE framework proposed by He et al. [2].[2]
- **Stage 2: DINOv1 Self-Distillation.** In the second stage, we adopt DINOv1 as the self-distillation framework, which uses a momentum-updated teacher and multi-crop views, without the additional large-scale training or architectural modifications introduced in later DINO variants.
  - We build upon the DINO framework introduced by Caron et al. [1].[3]

Encoder Architecture and Variants. For the encoder architecture, we primarily adopt a ViT backbone with a patch size of 8 and an embedding dimension of 704, consisting of 12 transformer layers. This architecture is used consistently for both MAE pretraining and DINO fine-tuning, with the MAE decoder discarded after pretraining. Using a shared encoder design allows the MAE stage to provide a direct and stable initialization for DINO without architectural mismatch.

In addition to this primary configuration, we also experiment with a smaller ViT variant with reduced embedding dimensionality. This variant follows the same two-stage training procedure and is evaluated under the same downstream protocol.

2.2.2. DINO Baseline (Our Baseline Model)

As a comparison point to our two-stage MAE+DINO approach, we also evaluate a baseline model pretrained using the DINOv2 framework. DINOv2 extends the original DINO formulation by introducing improved training stability, enhanced data augmentation, and normalization strategies designed to scale self-supervised learning to larger datasets and higher-capacity backbones.

Following the method proposed by Oquab et al. [4].[4]we adopt the standard DINOv2 recipe but apply it to the same 96×96 training resolution and ViT backbones listed in Table 1 to ensure comparability across methods. The model is trained using a momentum-updated teacher network, multi-crop augmented views, and a self-distillation objective defined over prototype assignments. Unlike our MAE+DINO approach, DINOv2 does not include a reconstruction stage and relies solely on invariance-based pretraining.

For backbone architecture, we primarily use the ViT-P8-768 variant from Table 1, which serves as the strongest baseline configuration under our experimental setup. The training pipeline follows the official DINOv2 hyperparameters with minor adaptations for resolution and batch size constraints.

2.3. Model Architecture

	ViT-P8-704	ViT-P8-512	ViT-P8-768
Patch size	8	8	8
Embed dim	704	512	768
Depth	12	12	12
Heads	11	8	12
Methods	MAE+DINO	MAE+DINO	DINOv2
Role	Main	Variant	B

Table 1. Vision Transformer backbone configurations used across self-supervised methods.

All self-supervised methods are implemented using Vision Transformer (ViT) backbones with a fixed input resolution of 96×96. This resolution is directly from the input image dataset. Due to the nature of competition in this project, any change in input size can lead to a potential advantage, causing unfairness. We primarily use patch size of 8, which offers a finer-grained representation and aligns with the limited computational resources. In addition, we experiment with models with smaller capacity (fewer embedding dimensions).

The detailed model architectures are shown in Table 1.

Implementation Note. In our codebase[5], the first two backbone variants in Table 1 are referred to as mae_vit_base_patch8 and mae_vit_small_patch8. These names are retained for implementation consistency, but in this report we refer to the models using explicit architectural descriptors (e.g., ViT-P8-704 and ViT-P8-512) to avoid ambiguity with canonical ViT size conventions.

[2] Our modified implementation, adapted to the 96×96 input resolution, is available at https://github.com/HaojieYin517/MAE.

[3] Our adapted DINOv1 implementation, modified to load MAE-initialized encoders and our training configuration, is available at https://github.com/HaojieYin517/DINOv1.

[4] Our DINOv2 implementation using the official Facebook Research DINOv2 repository is available at https://github.com/YujiaZhang123/DINOv2_Base/.

[5] https://github.com/HaojieYin517/MAE/

2.4. Frozen Encoder for Downstream Tasks

After self-supervised pretraining, we extract and save an encoder-only checkpoint by removing the unnecessary heads naturally generated from DINO self-distillation, including teacher heads. By the end, the primary model ViT-P8-704 contains 71,718,592 (∼71.7m) parameters, satisfying the limitation of 100M parameters of project.

The clean encoder parameters are frozen and there are no gradient updates when extracting representation during downstream usage of evaluation. Downstream performance is assessed by k-NN classification and linear probing with multinomial logistic regression.

3. Experiments

3.1. Datasets

3.1.1. Self-Supervised Pretraining Dataset

Encoder is trained on the Fall2025 Deep Learning Challenge dataset [3], which contains 500,000 unlabeled images. The dataset was created for an in-class self-supervised learning challenge and provides diverse visual content suitable for learning general-purpose image representations.

3.1.2. Downstream Evaluation Datasets

Downstream evaluation is conducted on three standard labeled image classification datasets provided: CUB-200-2011, Mini-ImageNet, and SUN397. Those datasets are preprocessed by the course staff, resizing images into 96 × 96 (align with pretraining images input) and splitting into fixed train, validation, and test sets. All evaluation datasets are used only in downstream evaluation, and no data from those datasets is seen during the self-supervised pretraining step, avoiding data leakage.

3.2. Training Details

MAE + DINOv1.

- All models are trained from scratch using the AdamW optimizer.
- During MAE, input images are normalized using channel-wise mean and standard deviation computed from the pretraining dataset. These statistics are fixed after pretraining and reused during downstream evaluation.
- For MAE pretraining, we use an effective batch size of 1280 and train the model for up to 400 epochs with a base learning rate of 3×10^{-4} and a linear warmup schedule.
- A high masking ratio of 0.85 is adopted to encourage learning from sparse visible patches.
- Following MAE pretraining, the encoder initializes the DINOv1 student, which is trained for 200 epochs using a momentum-based teacher.
- The DINOv1 teacher momentum is set to 0.997 to stabilize representation updates for large-capacity models.

DINO baselines.

- DINOv2 baselines are trained using 2 global crops and 6 local crops of size 48×48.
- Training runs for 180 epochs with 15 warm-up epochs, batch size 800, learning rate 3.7×10^{-4}, weight decay 0.04, and drop path rate 0.08.
- The teacher momentum increases from 0.995 to 0.9995 over training.
- The teacher temperature increases from 0.04 to 0.07 during the first 30 epochs and remains fixed thereafter.

3.3. Evaluation Protocol

To evaluate the performance of pretrained encoders, we apply a frozen-encoder evaluation protocol. After pretraining, all encoder parameters are fixed and used for feature extraction only.

The evaluation pipeline is described as below:

1. We first normalize all images using the same channel-wise statistics computed from the self-supervised pretraining dataset.
2. For each image, we extract the cls token from the final transformer layer as features.
3. Using the features extracted as input and labels as target, we fit classifiers using k-nearest neighbors (k-NN) classification and linear probing with the training set.
   - **k-NN** We systematically evaluate combinations of the following arguments:
     - Number of neighbors $k \in \{1, 3, 5, 10, 20, 50, 100\}$.
     - Distance metrics: cosine similarity and Euclidean distance.
     - Neighbor weighting strategies: uniform weighting and distance-based weighting. All features are L2-normalized.
   - **Linear probing** We train a multinomial logistic regression classifier. No further normalization applied.
4. For each classification model fitted, we compute and compare the classification accuracy on the validation set. **Classification accuracy**, defined as the percentage of correctly classified images over the entire test set, is the final evaluation metric.
5. The model achieving the highest accuracy on the validation set is selected and used to generate predictions on the test set, which are stored in a CSV file.
6. This CSV file is then uploaded to the Kaggle evaluation server, where final performance is reported on both the public and private test splits. Both splits share similar data distributions and are used to assess generalization performance.

4. Results and Analysis

4.1. Evaluation Classifier Comparison

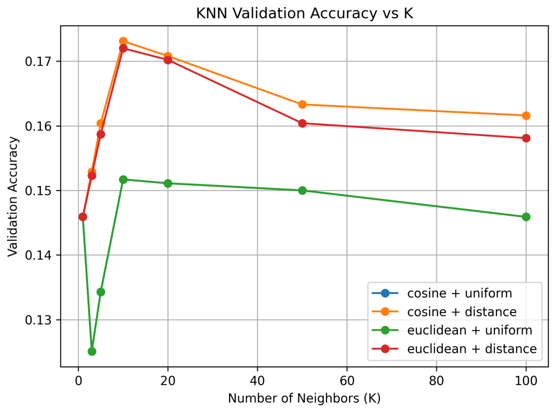

Figure 1. Validation accuracy of k-NN classifiers under different numbers of neighbors K, distance metrics, and weighting strategies. Results are shown for a representative evaluation dataset.

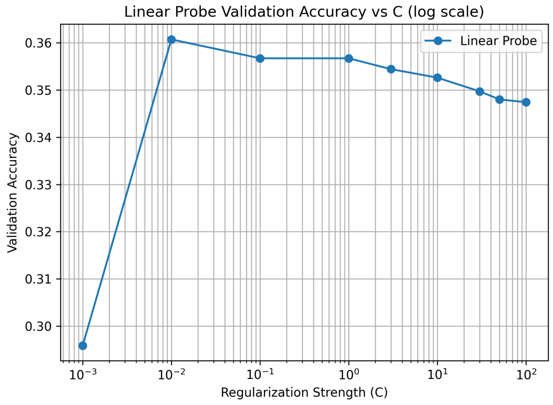

Figure 2. Validation accuracy of linear probing as a function of regularization strength C (log scale) for the same evaluation dataset.

We compare two frozen-encoder evaluation classifiers, k-nearest neighbors (k-NN) and linear probing. A representative example from Dataset 1 is shown in Figures 1 and 2, with consistent patterns observed across all evaluation datasets.

Overall, k-NN achieves relatively low validation accuracy across a wide range of neighborhood sizes, distance metrics, and weighting strategies, with moderate values of K (around 10) performing best. In contrast, linear probing consistently yields substantially higher validation accuracy, with optimal performance at moderate regularization strengths (around $C = 10^{-2}$), indicating a favorable bias–variance trade-off. This performance gap is also observed on the test set; therefore, we use the best-performing linear probe classifier to generate all final test submissions.

4.2. Overall Performance Comparison

Table 2. Overall downstream classification accuracy on public and private test splits.

Model	Dataset	Public	Private
MAE+DINOv1 (Submission)	1	**0.400**	**0.373**
	2	**0.769**	**0.776**
	3	**0.502**	**0.496**
DINOv2 (Baseline)	1	0.362	0.329
	2	0.743	0.741
	3	0.478	0.482

Table 2 summarizes downstream classification accuracy on all public and private test splits. Across all three evaluation datasets, the ViT-P8-704 model trained by MAE and DINOv1 pipeline consistently outperforms the pure DINOv2 baseline. Although this comparison is not a controlled ablation with DINOv1, it suggests that MAE provides a useful initialization for self-distillation under limited computational resources.

In addition to higher peak accuracy, the hybrid MAE and DINOv1 model exhibits more consistent performance across public and private splits, indicating improved generalization rather than overfitting to a specific test subset.

DINOv3 [5] performs relatively poorly on Dataset 1. We interpret this behavior as a result of DINOv3's higher model complexity, which may not be well matched to the relatively small pretraining dataset of 500,000 images. Therefore, it is not evaluated on the remaining datasets to save computational resources. Given its reliance on long training schedules, we prioritize MAE-based initialization and moderate pretraining durations, allocating computational resources to alternative model configurations instead.

5. Conclusion

In this project, we experiment with a two-stage self-supervised learning pipeline that combines masked autoencoding and DINO-based self-distillation sequentially. Across all three evaluation datasets, the proposed MAE $\rightarrow$ DINOv1 approach consistently outperforms strong DINO baselines, achieving higher and more stable downstream classification accuracy. Our results suggest that MAE provides an effective initialization and a good starting point for self-distillation. These findings highlight that combining different self-supervised learning methods can potentially lead to unexpected positive interaction, offering better learning performance for encoders.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[2] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021. 2

[3] NYU DS-1008. Fall 2025 deep learning challenge dataset. https://huggingface.co/datasets/tsbpp/fall2025_deeplearning, 2025. In-class challenge dataset for self-supervised learning. 3

[4] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2

[5] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. 4