From DINO to DYNO

Increasing the Performance of DINOv2 with Limited Training Data Through Dynamic Augmentation Scheduling

Ryan Henry
Harvard University
Cambridge, MA
ryanhenry@college.harvard.edu

Alexander Pratt
Brown University
Providence, RI
alexander_pratt@brown.edu

Prazul Wokhlu
Harvard University
Cambridge, MA
prazulwokhlu@college.harvard.edu

***Abstract*—Training contrastive learning models with image augmentations is critical to allow for downstream image classification to be augmentation-invariant. Until now, dynamic augmentation scheduling for contrastive learning has not been widely investigated. In this paper, we present *DYNO*, a loss-based dynamic augmentation scheduler. When used for linear probing DINOv2 for a facial recognition task, *DYNO* leads to a 46.40% increase in classification accuracy compared to no augmentation scheduling and a 0.42% increase in comparison to current state-of-the-art dynamic augmentation scheduling methods when trained on five images. We prove that *DYNO* outperforms any other strength computation strategy while held at a constant amount of training images. As of today, *DYNO* is both the most data efficient and reliable dynamic scheduling method for linear probing DINOv2 for computer vision tasks.**

I. Introduction

Since its introduction, Self-Distillation with No Labels [1] and its iterations [2] have proven to be a quality backbone for many computer vision tasks. (Oquab et al., 2024) claim that using DINOv2 as a backbone shows very strong prediction capabilities without additional tuning. This is provable with ample training data, but the performance is not as clear with a small quantity of training data. Using 7% of the DigiFace-1M dataset for linear probing, DINOv2 has 58.30% test accuracy for the facial recognition task.

Image augmentation is necessary for effective contrastive learning. Without augmentation during training, there is potential that augmentations in inference inputs may lead to incorrect classification [3], [4]. Data augmentation has been studied widely [4], [6], [7], [9], but there has been limited research into using dynamic augmentation scheduling for contrastive learning. (Zhang et al., 2023) have shown initial attempts to use dynamic augmentation scheduling for contrastive learning, but their research can further expanded upon.

We are motivated to create a novel dynamic augmentation scheduling system with a goal to increase the performance of linear probing DINOv2 using limited training data and to outperform current state-of-the-art augmentation strategies. Our work focuses on the facial recognition task, but we hope to prove that these trends are applicable for all tasks that have minimal training data. In total, our work includes:

- *DYNO*, a loss-based dynamic augmentation scheduler that uses novel techniques such as *strength scaling* to maximize performance for linear probing DINOv2 with a constrained amount of labeled images.
- In comparison to no or static augmentation, increased inference accuracy for the facial recognition task using *DYNO* with minimal training data.

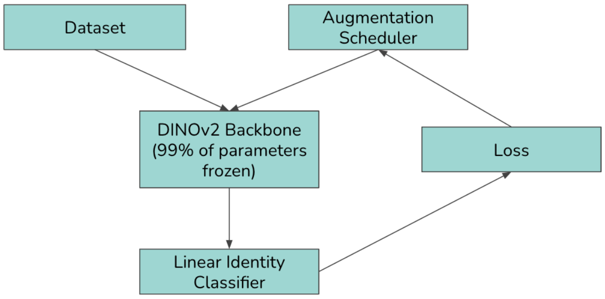

Dataset
Augmentation Scheduler
DINOv2 Backbone (99% of parameters frozen)
Loss
Linear Identity Classifier

Fig. 1: A diagram of the DYNO process. The dataset is paired with the augmentation strength to train DINOv2 with an identity classifier. The resulting loss is used to compute the next augmentation strength.

II. Approach

Current research in dynamic augmentation scheduling for contrastive learning uses a decreasing, piecewise approach to determine how augmentation should be applied for each epoch of training [5]. Our approach includes testing various methods that may have an impact on classification accuracy using our augmentation schedule including: strength computation strategies, direction of strength change, and scaling the augmentation strength to maximize augmentation.

Augmentation Schedule. The augmentation schedule is composed of six sections, each of which applies varying augmentation features to training images. (Chen et al., 2020), (Shorten, C. et al., 2019) present effective augmentation features for contrastive learning, which we use to select the features for each section. As the section strength increases, so does the intensity of the augmentation within each section. As a result, Random Resized Crops in the Weak section will have a weaker crop than the same feature in the Very Strong section. Information about each section is in Table I.

TABLE I: Dynamic Augmentation Scheduler Sections

Augmentation Section	**Table Column Head**	
	Strength Range	***Augmentation Features***
None	s = 0.0	No augmentation
Very Weak	$s < 0.20$	Weak Random Resized Crop
Weak	$s < 0.40$	Weak Random Resized Crop Random Horizontal Flip
Moderate	$s < 0.60$	Random Resized Crop Random Horizontal Flip Weak Color Jitter
Strong	$s < 0.80$	Strong Random Resized Crop Random Horizontal Flip Color Jitter
Very Strong	$s < 1.0$	Strong Random Resized Crop Random Horizontal Flip Strong Color Jitter Random Rotation

Strength Computation Strategies. Determining the strength of augmentation is vital for guiding a dynamic augmentation schedule. We present novel strength computation strategies to analyze the performance of dynamic augmentation scheduling. To design *DYNO*, analysis was performed on simple approaches including Epoch, Loss, and Validation Accuracy, and compound approaches including Epoch and Loss, and, Epoch and Validation Accuracy.

In addition to our designed approaches, we use the computation described by (Zhang et al., 2023). The associated augmentation features used in this paper are not released, so we cannot wholistically compare *DYNO* to their initial implementation. However, using their scheduling strategy enables a direct comparison to determine the best scheduling strategy for our augmentation implementation.

Strength Change Direction. The literature contains examples of dynamic augmentation in a decreasing direction [5]. We test both increasing and decreasing methods to determine if increasing dynamic augmentation strength produces better results or if we can improve upon current work in the decreasing direction.

Inter-Range Strength Scaling. Non-linear strength scaling is applied in each augmentation section to make the intensity of the augmentation a function of the augmentation range and the strength. To the best of our knowledge, this is a novel approach. Equation (1) represents strength scaling, where α is the scaling factor that changes with augmentation intensity.

$$\beta = (1 - \alpha) + (\alpha * (1 - strength)) \tag{1}$$

III. Implementation

The design of *DYNO* relies on the DINOv2 model. After obtaining the pre-trained base model, 99% of its layers are frozen and the model is paired with an identity classifier. Linear probing begins with the DigiFace-1M dataset by training 1% of parameters on DINOv2 and the associated identity classifier for 100 epochs. In each epoch, a new augmentation strength is computed based on the selected strategy. This strength is utilized to determine the augmentation section and scaled strength of augmentation.

Labeled Data. *DYNO* is designed to increase the performance of linear probing DINOv2 with minimal training data. As a result, all experiments use approximately 7% (5 images) or 14% (10 images) of available data per subject for labeled data. Initial training was performed with 20% of available data. These results offer little room for growth since no augmentation during training achieved 96.89% and *DYNO* achieved 97.26% test accuracy. This is a result of DINOv2 being extremely accurate with ample training data.

Validation Data. Some strength computation strategies require validation data. For a constrained amount of labeled data, these approaches require that less data is used for training. Comparing approaches that do and do not use validation data provides insight into what strength computation strategy results in the best classification accuracy and is most efficient with a constrained number of labeled images. Dataset splits in this paper are defined as A:B:C, where A is the number of training images, B is the number of validation images, and C is the number of test images.

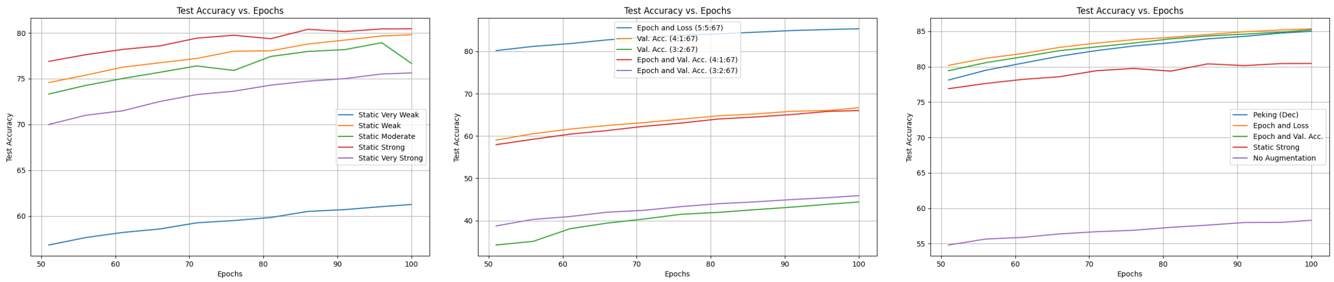
Static Experiments. Testing a linear probed DINOv2 model for the facial recognition task with no augmentation resulted in 58.30% test accuracy when trained on 7% of the dataset. Even if dynamic augmentation creates improvement in test accuracy compared to this result, there is potential that a static augmentation approach could create better results. To prove that our dynamic scheduler is the best method to improve accuracy with minimal training data, all augmentation strength sections were statically tested.

Approach Considerations. Where applicable, exhaustive combinations of validation data split and direction were tested for each strength computation strategy. Obtaining data for these combinations enables a wholistic comparison to determine the best scheduling scheme between static, the current state-of-the-art, and our dynamic scheduling strategies.

Development Information. All code of *DYNO* is written in Python and hosted on Google Colab. Libraries including NumPy and Parameter-Efficient Fine-Tuning (PEFT) [8] are used for optimized computation and to freeze the trained base model. The backbone used for testing is DINOv2. The associated dataset with this work is the DigiFace-1M dataset. This dataset contains 2000 subjects, each with 72 images. All subjects have source images at different ages, lighting, and clothing. This dataset was selected to prevent training on actual humans to preserve privacy.

IV. Results

Comprehensive tests were performed to ensure that *DYNO* presents the highest accuracy compared to other scheduling techniques when linear probing DINOv2. With the intent of making a scheduler that is effective for minimal labeled data, *DYNO* uses the decreasing, Compound Epoch and Loss computation strategy since it outperformed all strength computation strategies with our augmentation schedule for a set



(a) Comparing Static Augmentation Scheduling

(b) Comparing Constant Amount of Labeled Data

(c) Overall Comparison

Fig. 2: Comparisons of different augmentation approaches. (a) visualizes testing all augmentation schedule sections statically at a 5:5:62 split. (b) is the plot comparing *DYNO* to validation data-based methods constrained to five labeled images. (c) contains the best results of all categories at for five training images. All plots show results for epochs 50 to 100 for brevity.

amount of training data and requires no labeled images for validation. The strength computation strategy for *DYNO* is (2), where α and β are scalar hyper-parameters and σ represents strength for each respective type.

$$\sigma_{DYNO} = min(1, \alpha * \sigma_{epoch} + \beta * \sigma_{loss}) \qquad (2)$$

Static Augmentation. Static testing was completed at a 5:5:62 split for each augmentation strength range. The Strong range performed best at 80.45% test accuracy and the Very Weak range performed worst at 61.02% test accuracy. The first result demonstrates that with adequate augmentation, there can be increased test accuracy. The latter result resembles the quality of the DINOv2 model with no augmentation, which is explained by the Very Weak section having little augmentation.

Impact of Direction. All strength computation strategies were tested using a 5:5:62 split in an increasing and decreasing direction. As seen in Fig. 3, regardless of the strength computation strategy, the decreasing direction always has a higher test accuracy. As a result, decreasing approaches were the only approaches investigated for the remainder of testing in this body of work.

Better performance in the decreasing strategies is explained by the impact of early aggressiveness in augmentation. Starting with weak augmentation will not prepare the model for harsh augmentation. When a new augmentation section is entered for increasing strategies, there is a spike in loss in the training process. Early aggressive augmentation reduces the potential for weaker augmentation to shock the training process.

Impact of Loss-Based Approach. The Compound Epoch and Loss computation strategy is the highest performing dynamic augmentation scheduling strategy with a test accuracy of 85.35% after 100 epochs of training on a 5:0:67 split. Comparing to other augmentation strategies, *DYNO* has increased performance of no augmentation by 46.40%, the best static augmentation method by 6.09%, and the current state-of-the-art computation strategy by 0.42%.

Holding labeled data constant at 7% of available data, *DYNO* outperforms all methods that require validation data. The best strength computation strategies at a 4:1:67 split achieved 66.69% and at a 3:2:67 split achieved 45.9% test accuracy, resulting in an increase in accuracy of 27.98% and 85.95%, respectively. This demonstrates that with a limited amount of images for labeling, validation-based computation strategies are inefficient since they require taking data away from the training set.

Holding the amount of training data constant at 7%, validation-based methods at a 5:5:62 split are able to surpass all methods but *DYNO* with 85.19% test accuracy. With ample source images for labeling, this could be an alternative computation strategy for dynamic scheduling. However, since DINOv2 is effective with ample training data, the use of ten training images without any augmentation may have better accuracy, implying any augmentation is almost unnecessary.

V. Related Work

AutoAugment. (Cubuk et al., 2019) present work that automates the search for a generalized, effective static augmentation policy by optimizing combinations of transformations for consistent performance gains. *AutoAugment* does not investigate dynamic augmentation, but it does offer insight into choosing the most effective static augmentation that can be used in a broader application.

DYNACL. (Zhang et al., 2023) demonstrate that they can produce significant improvements in performance by using a decreasing dynamic augmentation scheduler for training contrastive learning models. We consider this work the state-of-the-art when comparing our scheduler. Our approaches differ based on how we compute our strength in (2) and how we apply the strength scaling factor in (1) to have non-linear, increasing augmentation strength within each augmentation section. Additionally, we are testing our work using linear probing, while *DYNACL* was tested by training a model.

FRoundation. In *FRoundation*, (Chettaoui et al., 2024) investigates the use of foundation models and their potential use for the facial recognition task. This implementation relies on static augmentation during the training process, and fine-tunes the foundation models with task-specific data. In our implementation, we use dynamic augmentation during the

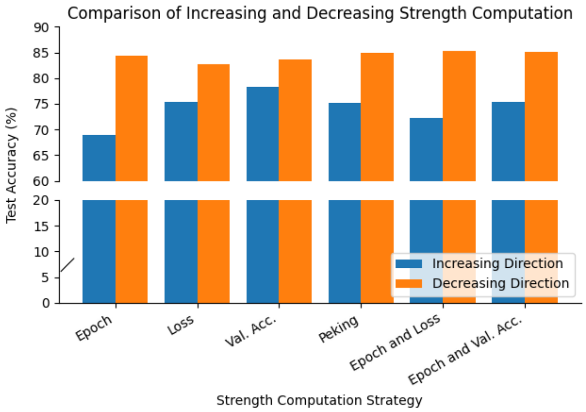


Fig. 3: A comparison of final test accuracies of all strength computation strategies in an increasing and decreasing direction

training process, and we are linear probing, not fine-tuning, DINOv2.

VI. Conclusion

Since there is minimal research into dynamic augmentation for training contrastive learning models, this work can contribute useful findings to the literature. In summary, we assert the following:

- Compared to increasing, decreasing dynamic augmentation strategies produce higher classification accuracy for a computer vision task using DINOv2.
- Dynamic augmentation improves test accuracy compared to no or static augmentation for minimal labeled data.
- Scaling the strength within an augmentation section increases classification accuracy.
- Using both current epoch and loss to compute the augmentation strength on a dynamic scheduler produces the highest classification results for minimal labeled data.
- Validation-based strategies are wasteful since they require roughly twice the amount of labeled images to achieve near-performance as other strategies.

Our project goal was to determine if we could use dynamic augmentation to increase classification accuracy for linear probing a contrastive learning model. Based on the results obtained and the contributions identified in this section, we believe that we achieved our goal. There can be further improvements and investigations for this work, which are highlighted in Future Work.

Future Work

Varying Datasets. This work centers around the use of the DigiFace-1M dataset. This data is a great starting point for analyzing the efficacy of our implementation. To further prove the performance of *DYNO*, it is important to attempt linear probing DINOv2 with other datasets for different tasks. This analysis can help prove the concept that dynamic augmentation scheduling is useful for all tasks that use DINOv2 as a backbone with limited training data.

Range Biasing. We attempted to analyze the impact of using an approximate mapping function to bias the computed strength toward the most effective static augmentation section. The implementation used three approaches: a cubic function, a sinusoidal function, and a quadratic sinusoidal function. Some of the results proved to be promising, but there was not enough time to complete a full analysis of the impacts. Fig. 1 shows the mapping functions and how they relate to the unmapped approach.

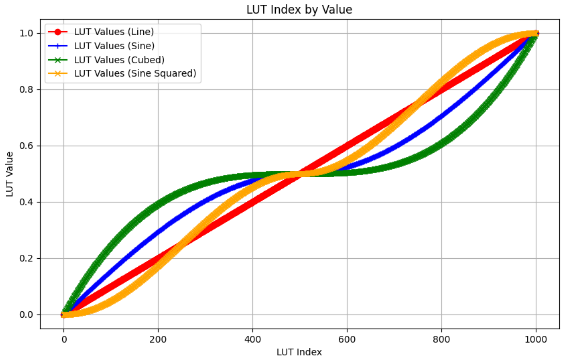


Fig. 4: Representation of Attempted Mapping Biases

Augmentation Features. When creating the scheme, we used [3], [4] to select augmentation features. There is potential that augmentations sections could be modified or changed to add inherently more or less aggressive augmentation features to evaluate the performance of *DYNO*. Although augmentation strength is used to make each augmentation more or less aggressive, identifying the optimal augmentation features could prove to add benefit to the system.

Most of the dynamic decreasing strength computation strategies resulted in similar test accuracies. There is potential that the augmentation features contributed to the increased accuracy, and not the scheduling strategy. Examining the augmentation features selected could help prove whether this is true or not.

We do not know the augmentation features used by [5]. As a result, our comparison to their paper was based on their computation strategy with our augmentation schedule. The similarity in performance between our two methods could be a result of the augmentation features in our schedule, and not because their overall approach is similar. It would be good to compare our scheduler and features with their initial implementation to verify the difference in performance.

Strength Scaling Factor. The scaling factor used for (1), α, was chosen based on a need to limit the scaling range between 0 and 1, without making the lower bound 0. This is because a lower bound of 0 could potentially lead to too aggressive augmentation for the Very Strong section. In this paper, we do not disclose the actual range we used. Future work can include a grid search to determine the optimal scaling factor range.

Individual Contribution

All three members significantly contributed to the project by assisting with all testing, preparing presentation materials, and developing the approach to be implemented. Below lists specific contributions from each team member.

Ryan Henry. Ryan led the effort to implement the framework used for training and testing our implementation. This included choosing the augmentation features used in the schedule, considering different strength computation strategies, and designing the interface of the augmentation scheduler with the training mechanism.

Alexander Pratt. Alec contributed to the design and implementation of the augmentation scheduler, specifically the novel strength scaling mechanism and strength computation strategies. In addition, Alec worked on implementing and testing differing bias techniques that are mentioned in Future Work. Alec wrote the written report for the project.

Prazul Wokhlu. Prazul assisted in the development of the framework of our implementation including freezing the DINOv2 model using PEFT, combining the frozen model with a linear classifier, and designing the software used for training the system. Prazul helped with aspects of the report, and with data analysis and visualization.

Acknowledgment

This project was completed as part of requirements for COMPSCI 2420: Computing at Scale, at Harvard University in the Fall 2024 semester. Thank you to Dr. H.T. Kung, for his instruction in lecture through the semester and helpful feedback for the direction of this project. Additionally, thank you to the entire teaching staff for their support in completing this project.

References

[1] Caron, M., et al. ”Emerging Properties in Self-Supervised Vision Transformers.” Advances in Neural Information Processing Systems (NeurIPS), 2021.

[2] Oquab, M., et al. ”DINOv2: Learning Robust Visual Features without Supervision.” Transactions on Machine Learning Research, 2024.

[3] Chen, T., et al. ”A Simple Framework for Contrastive Learning of Visual Representations.” International Conference on Machine Learning (ICML), 2020.

[4] Shorten, C., and Khoshgoftaar, T. M. ”A Survey on Image Data Augmentation for Deep Learning.” Journal of Big Data, 2019.

[5] Zhang, C., et al. ”Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning.” International Conference on Learning Representations (ICLR), 2023.

[6] Cubuk, E. D., et al. ”AutoAugment: Learning Augmentation Policies from Data.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[7] DeVries, T., and Taylor, G. W. ”Improved Regularization of Convolutional Neural Networks with Cutout.” arXiv preprint arXiv:1708.04552, 2017.

[8] Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. ”Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey.” unpublished, 2024.

[9] Chettaoui, T., Damer, N., and Boutros, F. ”FRoundation: Are foundation models ready for face recognition?” unpublished, 2024.