On the Importance of Hyperparameters and Data Augmentation for Self-Supervised Learning

Diane Wagner [1] **Fabio Ferreira** [1] **Danny Stoll** [1] **Robin Tibor Schirrmeister** [1] **Samuel Müller** [1] **Frank Hutter** [1 2]

Abstract

Self-Supervised Learning (SSL) has become a very active area of Deep Learning research where it is heavily used as a pre-training method for classification and other tasks. However, the rapid pace of advancements in this area comes at a price: training pipelines vary significantly across papers, which presents a potentially crucial confounding factor. Here, we show that, indeed, the choice of hyperparameters and data augmentation strategies can have a dramatic impact on performance. To shed light on these neglected factors and help maximize the power of SSL, we hyperparameterize these components and optimize them with Bayesian optimization, showing improvements across multiple datasets for the SimSiam SSL approach. Realizing the importance of data augmentations for SSL, we also introduce a new automated data augmentation algorithm, *GroupAugment*, which considers groups of augmentations and optimizes the sampling across groups. In contrast to algorithms designed for supervised learning, GroupAugment achieved consistently high linear evaluation accuracy across all datasets we considered. Overall, our results indicate the importance and likely underestimated role of data augmentation for SSL.

1. Introduction

Self-supervised learning (SSL) has seen an explosion of research interest in recent years, with significant progress using SSL as a pre-training method for classification (Grill et al., 2020; Chen & He, 2021; Chen et al., 2020a; He et al., 2020; Chen et al., 2020b). A large variety of SSL methods have been developed, for example using different optimization paradigms (van den Oord et al., 2018; Pathak et al., 2016), different objective functions (Giradis et al., 2018; Doersch et al., 2015; Zhang et al., 2016), or different data modalities (Radford et al., 2021).

An aspect of SSL performance that is less researched is the effect of other choices, such as the training hyperparameters or the augmentation strategy. To shed light on these neglected factors, we use Bayesian optimization (Mockus et al., 1978; Shahriari et al., 2016) to search for configurations of the SimSiam SSL algorithm (Chen & He, 2021) on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and the medical dataset DermaMNIST (Yang et al., 2021a;b). We consider, on the one hand, a search over training hyperparameters and, on the other hand, a search over data augmentation strategies. Among other findings, our results suggest the importance of data augmentation for SSL.

Motivated by the apparent importance of data augmentation for SSL, we then develop a new automated data augmentation algorithm, *GroupAugment*, that covers a more diverse space of augmentation strategies than existing methods and can, e.g., design augmentation strategies that resemble manually-designed SSL augmentation strategies.

In summary, our main contributions are:

- We study the effect of training hyperparameters and augmentation strategies for the SimSiam SSL approach (Section 3). Our results indicate the importance of data augmentation for SSL.
- We introduce the automated data augmentation algorithm GroupAugment and demonstrate its high performance for SSL (Section 4).

2. Background and Related Work

Self-Supervised Learning The most cited works in Self-Supervised Learning, such as SimCLR, BYOL, SimSiam, MoCo, or DINO (Grill et al., 2020; Chen & He, 2021; Chen et al., 2020a; He et al., 2020; Chen et al., 2020b) all apply a similar or identical data augmentation protocol (random horizontal flip, color distortion, and nowadays also Gaus-

[1]Department of Computer Science, University of Freiburg, Freiburg, Germany [2]Bosch Center for Artificial Intelligence, Germany. Correspondence to: Diane Wagner <wagnerd@cs.uni-freiburg.de>.

First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

sian blur and solarization). Most relevant to our work are the findings of Grill et al. (2020) and Chen et al. (2020a), who identify and address the sensitivity to choosing color distortions in their methods. Moreover, Chen et al. (2020a) identified that a sophisticated supervised data augmentation strategy does not perform better than simple cropping with strong color distortion in the self-supervised setting. These observations motivate the contributions regarding data augmentation for SSL, which we will revisit next.

Data Augmentation Algorithms In supervised learning, data augmentation algorithms usually outperform manually selected augmentation strategies: Three algorithms that rely on randomly sampling augmentations from a fixed set of augmentations are TrivialAugment (Müller & Hutter, 2021) and SmartSamplingAugment (Negassi et al., 2022). Other algorithms, such as AutoAugment (Cubuk et al., 2019), RandAugment (Cubuk et al., 2020), and SmartAugment (Negassi et al., 2022) perform a gradient-free search in the space of augmentation policies. It is also possible to apply gradient-based optimization to meta-learn pre-training hyperparameters (Raghu et al., 2021) for ECG data. In SelfAugment (Reed et al., 2021), the authors bootstrap from the correlation between supervised and self-supervised evaluation performance to incorporate a rotation task for generating augmentation policies efficiently. SelfAugment is qualitatively different from the previous approaches in that it does not optimize for general downstream performance but rotation task performance. Additionally, it will likely not include rotations into the selected augmentation policy.

In contrast to these, our approach *GroupAugment* (Section 4) covers a more diverse space of augmentation strategies than existing methods and can, e.g., design augmentation strategies that resemble manually-designed SSL augmentation strategies. It generalizes existing approaches by optimizing group-specific sampling probabilities, the number of group-specific augmentations, and the total number of augmentations applied while imposing no limitations on the set of augmentations such as SelfAugment.

3. Study on the Importance of Hyperparameters and Data Augmentation

We study the following research questions:

- What role does data augmentation play in SSL, and can better data augmentation strategies lead to better performance?
- Which hyperparameters may be notorious for resulting in model collapse when set incorrectly?
- Which hyperparameters are important to optimize in SSL to achieve good performance and outperform baselines?

3.1. Study Design

To answer the presented questions, we conducted the study described below.

Models, Datasets and Hyperparameter Search Spaces We perform all our experiments on the CIFAR-10, CIFAR-100 and DermaMNIST datasets (Krizhevsky, 2009; Yang et al., 2021a;b) and use the SimSiam (Chen & He, 2021) approach with the ResNet-18 (He et al., 2016) architecture. We optimize a wide range of training pipeline hyperparameters (which we refer to as *Training Hypers*) such as the learning rate, warmup and weight decay, and optimizer, as well as data augmentation hyperparameters (which we refer to as *Augmentations*) involving magnitudes and probabilities of image distortions. Please see Appendix A for more details on our search spaces. We point out that all hyperparameter search spaces of the baselines are chosen identically across all datasets except for the pre-training epochs, where we used 800 for CIFAR-10/100 and 100 for DermaMNIST. Lastly, we report how the train, validation, and test splits were chosen in Appendix B.

Search Algorithm To optimize over the search spaces listed above, we use Bayesian optimization (BO) (Mockus et al., 1978) with expert priors (Hvarfner et al., 2021) as implemented by Stoll et al. (2022). For details on the chosen priors see Appendix B.

Performance Evaluation All reported performance values are based on the standard linear evaluation protocol (Dalal & Triggs, 2005; Grill et al., 2020) from the SSL literature that trains a linear classifier on top of the frozen ResNet backbone weights.

3.2. Results

Hyperparameters vs Data Augmentation Strategy Table 1 shows that (a) optimizing six training hyperparameters (detailed in Table 5 in the appendix) of SimSiam only lead to marginal improvements or even deterioration of performance (due to differences in validation and test split); (b) optimizing the data augmentation strategy lead to consistent significant performance improvements (at least 1%, and up to 2.3% for CIFAR-100). This shows that SimSiam's training hyperparameters were already very well-tuned.

How to Avoid Collapsing Chen & He (2021) already analyzed which factors, e.g., stop-gradient, can cause collapsing solutions for SimSiam. A collapsing solution is an undesired solution where all the outputs collapse to a constant vector. We continue this study and give insights into which hyperparameters may cause collapsing if chosen suboptimally. While we observed collapsing solutions in some of our experiments on the CIFAR-10 and CIFAR-100

Table 1. Mean test accuracy [%] for SimSiam and its tuned variants in the linear evaluation protocol. We report the mean and standard error across five seeds. (†) The original SimSiam result of 91.8% was achieved using early stopping on the test set.

Approach	DermaMNIST	CIFAR-10	CIFAR-100
SimSiam (Chen & He, 2021)	66.2 ±0.3	91.6 ±0.1†	65.6 ±0.2
SimSiam Tuned Training Hypers	66.5 ±0.1	91.6 ±0.1	64.9 ±0.2
SimSiam Tuned Augmentations	**67.2** ±0.4	**92.7** ±0.1	**67.9** ±0.3

datasets, for DermaMNIST, we surprisingly observed no collapsing solutions. We show violin plots in Appendix D giving some insights on which hyperparameters may be responsible for collapsing solutions. For example, Figure 1, an excerpt from Appendix D, indicates that a probability of applying grayscale to CIFAR-100 around 0.5 might cause collapsing solutions.

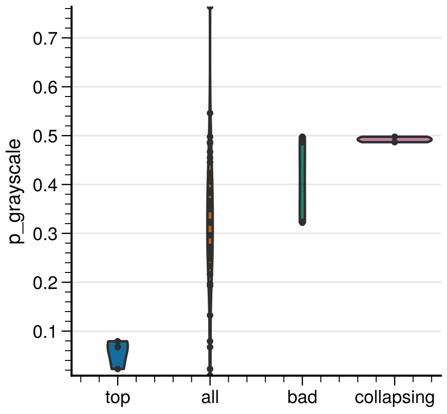

Figure 1. Density estimates for the probability of applying a grayscale augmentation as sampled in our optimization for SimSiam's augmentation strategy on CIFAR-100. We show density estimates for the best and worst-performing configurations (top and bad), all configurations (all), and configurations leading to collapsing.

Importance of Hyperparameters In the following, we study the importance of individual hyperparameters for achieving good performance. First, we compare the manually designed hyperparameter settings from Chen & He (2021) to the optimized settings found in our study. As Chen & He (2021) do not report results on CIFAR-100 and DermaMNIST, we focus on CIFAR-10 here. For training hyperparameters, we do not observe significant differences in performance (cf. Table 1), and the settings found by (Chen & He, 2021) seem to be quite optimal. For the parameterized data augmentation config space, we observe that the grayscale probability hyperparameter should be set half as high as in the baseline and that the saturation strength from the color jitter needs to be higher than in the baseline. Further, adding solarize seems important, which is also observed by Grill et al. (2020). The hyperparameter importance study in Table 2 also supports the importance of the above parameters. Additionally, Figure 2 suggests that for the above parameters, good and bad configurations differ, supporting our findings. Moreover, our results also give insights into how individual hyperparameters should be optimally set. As an example, Figure 1 shows how the probability of applying grayscale should be optimally set for the CIFAR-100 dataset. In the best-performing configurations, this probability hyperparameter has been sampled below 0.1 and in bad-performing configurations above 0.3. We show violin plots of all the other individual hyperparameters of the different config spaces and datasets in Appendix D.

Table 2. Hyperparameter importances for the SimSiam data augmentation space for CIFAR-10. We utilized fANOVA (Hutter et al., 2014), which quantifies the contribution of individual hyperparameters to the overall variance in performance. We distinguish between the general importance and the importance of best configurations. Higher importance values denote a higher performance responsibility.

Hyperparameter	Across all	Across best
brightness strength	1	2
contrast strength	2	2
hue strength	8	3
p colorjitter	2	2
p grayscale	16	32
p horizontal flip	8	2
p solarize	30	26
saturation strength	23	7
solarize threshold	5	2

4. GroupAugment

In this section, we introduce *GroupAugment*, an automated data augmentation algorithm that operates on groups of augmentations (such as color or quality transformations) and designs sampling strategies over these groups. Further, we present an empirical study where we find that, in contrast to the data augmentation algorithms designed for supervised learning, GroupAugment robustly outperforms the baseline in all the settings we analyzed.

Table 3. Mean test accuracy for different algorithms in the linear evaluation protocol. We report the mean and standard error across five seeds for methods we ran. For each dataset, we bold the two best accuracies and underline scores that outperform the SimSiam baseline. (†) The original SimSiam result of 91.8% was achieved using early stopping on the test set. (‡) SelfAugment and SelfRandAugment were evaluated using Resnet50 and not Resnet18 as other methods. Reed et al. (2021) report results for multiple instantiations of SelfAugment.

Approach	DermaMNIST	CIFAR-10	CIFAR-100
SimSiam (Chen & He, 2021)	66.2 $\pm$0.3	91.6 $\pm$0.1†	65.6 $\pm$0.2
SelfRandAugment (Reed et al., 2021)	-	90.3‡	-
SelfAugment (Reed et al., 2021)	-	$87.5 - 92.6^\ddagger$	-
RandAugment (Cubuk et al., 2020)	<u>**68.8**</u> $\pm$0.3	89.9 $\pm$ 0.0	59.3 $\pm$ 0.5
SmartAugment (Negassi et al., 2022)	<u>67.5</u> $\pm$0.1	89.8 $\pm$ 0.1	59.8 $\pm$ 0.1
TrivialAugment (Müller & Hutter, 2021)	<u>67.7</u> $\pm$0.5	89.4 $\pm$ 0.1	59.1 $\pm$ 0.2
Tuned SimSiam Augmentations (our)	<u>67.2</u> $\pm$0.4	<u>**92.7**</u> $\pm$0.1	<u>**67.9**</u> $\pm$0.3
GroupAugment (our)	<u>**68.0**</u> $\pm$0.3	<u>**93.0**</u> $\pm$0.1	<u>**66.3**</u> $\pm$0.4

4.1. Algorithm

Augmentation Sampling Given some groups of data augmentations $\{g_i\}$, GroupAugment uses one global hyperparameter and two sets of group-specific hyperparameters to create a list of augmentation sequences that will be consecutively applied to an image. The global hyperparameter T determines the number of augmentation sequences in that list. The group-specific hyperparameter P_{g_i} determines the probability that augmentation group g_i is chosen to create the following augmentation sequence. N_{g_i} determines the number of augmentations to sample uniformly without replacement to form an augmentation sequence for augmentation group g_i. We provide pseudocode for a GroupAugment policy in Algorithm 1.

Algorithm 1 A GroupAugment policy applied to an image.

```
Input: Image $I$, augmentation groups $\{g_i\}$,
       group-specific sampling probabilities $\{P_{g_i}\}$,
       group-specific #augmentations $\{N_{g_i}\}$,
       total #group-samples $T$
Initialize empty list of augmentation sequences $A$
for augmentation sequence $1, \ldots, T$ do
  Sample group $g$ according to $\{P_{g_i}\}$
  Sample $N_g$ augmentations from group $g$
  Append augmentations to $A$
end for
Apply sampled augmentation sequences $A$ to $I$
```

Novelty of Group Sampling While Negassi et al. (2022) explored optimizing the parameters of color and geometric augmentation groups, GroupAugment generalizes this notion to *any* set of augmentation groups and searches over more general spaces of sampling strategies. In our study, we instantiate GroupAugment with five groups: color, geometric, non-rigid, quality, and exotic. See Table 9 in Appendix C for the specific augmentations we used and Table 6 in Appendix A for a detailed description of GroupAugment's search space.

Search Algorithm To optimize over the resulting search space, we use the same Bayesian optimization (BO) approach with expert priors as above. Further, we normalize the sampled probabilities, as BO samples each group probability individually, and therefore, they do not necessarily add up to 1.

4.2. Comparisons and Results

We compare GroupAugment to several automated data augmentation algorithms. On the one hand, we observe that applying standard automated data augmentation algorithms, e.g., RandAugment, to the self-supervised learning setting worsens the results for most of our analyzed datasets, although those algorithms perform well in the supervised-learning setting (see Table 3). This observation is in line with the finding of Chen et al. (2020a) that a sophisticated supervised data augmentation strategy did not perform better than simple cropping with strong color distortion in the SSL setting. On the other hand, contrary to standard supervised-learning data augmentation algorithms, GroupAugment robustly outperforms the baseline in all the settings we analyzed. We give more details on the experimental settings in Appendix B.

Further, we observe a performance improvement over the baseline when tuning its data augmentation strategy's magnitudes and application probabilities with the same computational budget we used for the GroupAugment search space. However, as GroupAugment outperforms the tuned SimSiam data augmentation for most of our results, allowing stronger parameterization, we recommend optimizing the data augmentation with GroupAugment, especially for datasets having no intuition about which data augmentation might be helpful.

5. Conclusion and Limitations

While SimCLR and BYOL have analyzed the role of data augmentation, our results show that it is beneficial to analyze it in much greater detail. In summary, we provide evidence for the underestimated role of data augmentation for SSL and present a novel automated data augmentation algorithm, GroupAugment, which outperforms vanilla-SimSiam across all datasets we study.

Limitations We conducted our study on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and DermaMNIST (Yang et al., 2021a;b). More datasets should be considered to gather more insights and strengthen our experimental conclusions. In particular, results on ImageNet (Deng et al., 2009) and more medical datasets are of interest. Further, as automated data augmentation (e.g., our GroupAugment) optimizes the augmentation strategy on a validation set, out-of-distribution test sets can pose a challenge if the validation set is in-distribution.

Acknowledgements

We acknowledge funding by Robert Bosch GmbH, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828, by European Research Council (ERC) Consolidator Grant "Deep Learning 2.0" (grant no. 101045765), and by BrainLinks-BrainTools which was funded by the German Research Foundation (DFG, grant no. EXC 1086) and is currently funded by the Federal Ministry of Economics, Science and Arts of Baden Württemberg within the sustainability program for projects of the excellence initiative. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the ERC can be held responsible for them.

Funded by
the European Union

References

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. Albumentations: Fast and flexible image augmentations. *Information*, 11 (2), 2020. ISSN 2078-2489. doi: 10.3390/info11020125. URL https://www.mdpi.com/2078-2489/11/2/125.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Cubuk, E., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Cubuk, E., Zoph, B., Shlens, J., and Le, Q. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *CVPR'05*, volume 1, pp. 886–893 vol. 1, 2005.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pp. 248–255, 2009.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the 18th International Conference on Computer Vision (ICCV'15)*, pp. 1422–1430, 2015.

Giradis, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Hutter, F., Hoos, H., and Leyton-Brown, K. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pp. 754–762. PMLR, 2014.

Hvarfner, C., Stoll, D., Souza, A., Lindauer, M., Hutter, F., and Nardi, L. πbo: Augmenting acquisition functions with user beliefs for bayesian optimization. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021. Published online: iclr.cc.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Mockus, J., Tiesis, V., and Zilinskas, A. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129), 1978.

Müller, S. G. and Hutter, F. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 774–782, October 2021.

Negassi, M., Wagner, D., and Reiterer, A. Smart(sampling)augment: Optimal and efficient data augmentation for semantic segmentation. *Algorithms*, 15(5), 2022. ISSN 1999-4893. doi: 10.3390/a15050165. URL https://www.mdpi.com/1999-4893/15/5/165.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Raghu, A., Lorraine, J., Kornblith, S., McDermott, M., and Duvenaud, D. K. Meta-learning to improve pre-training. *Advances in Neural Information Processing Systems*, 34, 2021.

Reed, C. J., Metzger, S., Srinivas, A., Darrell, T., and Keutzer, K. Selfaugment: Automatic augmentation policies for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2674–2683, 2021.

Shahriari, B., Swersky, K., Wang, Z., Adams, R., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Stoll, D., Schrodi, S., Janowski, M., Mallik, N., Théophane, V., and Hutter, F. Neural Pipeline Search (NEPS), May 2022. URL https://github.com/automl/neps.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Yang, J., Shi, R., and Ni, B. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021a.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021b.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *12th European Conference on Computer Vision (ECCV'16)*, pp. 649–666, 2016.

A. Search Spaces

Table 4. SimSiam data augmentation search space.

Hyperparameter	Type	Range	Log-Prior	Default
p_colorjitter	Float	[0, 1]	False	0.8
p_grayscale	Float	[0, 1]	False	0.2
p_horizontal_flip	Float	[0, 1]	False	0.5
p_solarize	Float	[0, 1]	False	0.2
brightness_strength	Float	[0, 1.5]	False	0.4
contrast_strength	Float	[0, 1.5]	False	0.4
saturation_strength	Float	[0, 1.5]	False	0.4
hue_strength	Float	[0, 0.5]	False	0.1
solarize_threshold	Integer	[0, 255]	False	127

Table 5. SimSiam training hyperparameters search space.

Hyperparameter	Type	Range	Log-Prior	Default
learning_rate	Float	[0.003, 0.3]	True	0.03
warmup_epochs	Integer	[0, 80]	False	0
warmup_multiplier	Float	[1.0, 3.0]	False	1.0
optimizer	Categorical	{AdamW, SGD, LARS}	False	SGD
weight_decay_start	Float	$[5 \cdot 10^{-6}, 5 \cdot 10^{-2}]$	True	$5 \cdot 10^{-4}$
weight_decay_end	Float	$[5 \cdot 10^{-6}, 5 \cdot 10^{-2}]$	True	$5 \cdot 10^{-4}$

Table 6. GroupAugment search space.

Hyperparameter	Type	Range	Log-Prior	Default
p_color_transformations	Float	[0, 1]	False	0.5
p_geometric_transformations	Float	[0, 1]	False	0.5
p_non_rigid_transformations	Float	[0, 1]	False	0.0
p_quality_transformations	Float	[0, 1]	False	0.0
p_exotic_transformations	Float	[0, 1]	False	0.0
num_color_transformations	Integer	[1, 5]	False	1
num_geometric_transformations	Integer	[1, 2]	False	1
num_non_rigid_transformations	Integer	[1, 3]	False	1
num_quality_transformations	Integer	[1, 2]	False	1
num_exotic_transformations	Integer	[1, 2]	False	1
num_total_group_samples	Integer	[1, 5]	False	1

Table 7. RandAugment search space.

Hyperparameter	Type	Range	Log-Prior	Default
num_ops	Integer	[1, 15]	False	3
magnitude	Integer	[0, 30]	False	4

Table 8. SmartAugment search space.

Hyperparameter	Type	Range	Log-Prior	Default
num_col_ops	Integer	[1, 9]	False	2
num_geo_ops	Integer	[1, 5]	False	1
col_magnitude	Integer	[0, 30]	False	4
geo_magnitude	Integer	[0, 30]	False	4
p_apply_ops	Float	[0, 1]	False	1

B. Experimental Details

B.1. Expert Priors

We use Bayesian optimization (BO) (Mockus et al., 1978) with expert priors (Hvarfner et al., 2021) as implemented in the NePS python package (Stoll et al., 2022). Therefore, we set expert priors to guide the search. The priors in NePS are, in the continuous case, Gaussian distributions centered at a default value with the standard deviation determined via a confidential setting. We always use a “medium” confidence and set the default value as described below.

Training Hyperparameters and Data Augmentation Strategy We set the defaults to the baseline for the training hyperparameters and data augmentation strategy. As no solarization is used in the SimSiam baseline and Chen & He (2021) shows that adding solarize might improve the performance, we add solarize to the search space and set the user prior default to the values reported by Chen & He (2021).

RandAugment For the RandAugment search space, we set the defaults according to the optimal data augmentation policy for CIFAR-10 following Cubuk et al. (2020).

SmartAugment As we are the first applying SmartAugment to classification, we set the user prior defaults based on the optimal data augmentation policy for CIFAR-10 following Cubuk et al. (2020).

GroupAugment For the GroupAugment config space, we set the default user priors to one augmentation per group. As only color and geometric augmentations occur in the baseline, we set the user prior defaults for these group probabilities to 0.5 and the other group probabilities user prior defaults to 0.

B.2. Resources and Compute Budget

For our Bayesian optimization runs, we allowed a budget of 50 evaluations. For CIFAR-10 and CIFAR-100, one configuration evaluation took $\approx$ 8h with one GeForce RTX 2080 Ti GPU, for DermaMNIST $\approx$ 10min with 1 GPU. We used 10 GPUs in parallel for CIFAR-10, 20 GPUs in parallel for CIFAR-100, and 5 GPUs in parallel for DermaMNIST. In order to take noise on the validation set during our HPO into account, we evaluate the best-performing configurations multiple times on the validation set.

B.3. GroupAugment Comparative Study: RandAugment Search Space

While SmartAugment and GroupAugment were designed with BO in mind, for RandAugment, which originally used Grid Search, we follow Negassi et al. (2022) and consider an extended search space for the number of operations between 1 and 15. Negassi et al. (2022) showed that a larger number of operations than 3 can be beneficial for the performance. See also Table 7.

B.4. Dataset Splits

Since CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) do not provide a validation set, we split the training set and randomly sample a fixed validation set containing 10% of the training data for our hyperparameter optimization. We use the entire training set for training for our final test evaluations of the best-performing validation configurations. For DermaMNIST (Yang et al., 2021a;b), we have adopted the provided training, validation, and test split.

C. GroupAugment Details

Table 9. Details concerning the data augmentations from the groups. In our implementation, we use the data augmentations from the albumentations library (Buslaev et al., 2020).

Group	Augmentation
color	ColorJitter(brightness=0.4, contrast=0.4, saturation=0.4, hue=0.1)
	ToGray()
	Solarize()
	Equalize()
	ChannelShuffle()
geometric	ShiftScaleRotate(interpolation=cv2.INTER_CUBIC)
	HorizontalFlip()
non-rigid	ElasticTransform(alpha=0.5, sigma=10, alpha_affine=5, interpolation=cv2.INTER_CUBIC)
	GridDistortion(interpolation=cv2.INTER_CUBIC)
	OpticalDistortion(distort_limit=0.5, shift_limit=0.5, interpolation=cv2.INTER_CUBIC)
quality	GaussianBlur()
	GaussNoise()
exotic	RandomGridShuffle()
	Cutout(num_holes=4)

D. Individual Hyperparameter Analysis

Here, we individually analyze each hyperparameter and augmentation strategy parameter considered in Section 3. In particular, we plot density estimates for the values sampled in our search. We show density estimates for the best and worst-performing configurations (top and bad), all configurations (all), and configurations leading to collapsing. For categorical values, we show the sample distribution.

D.1. Tuned SimSiam Data Augmentation Strategy

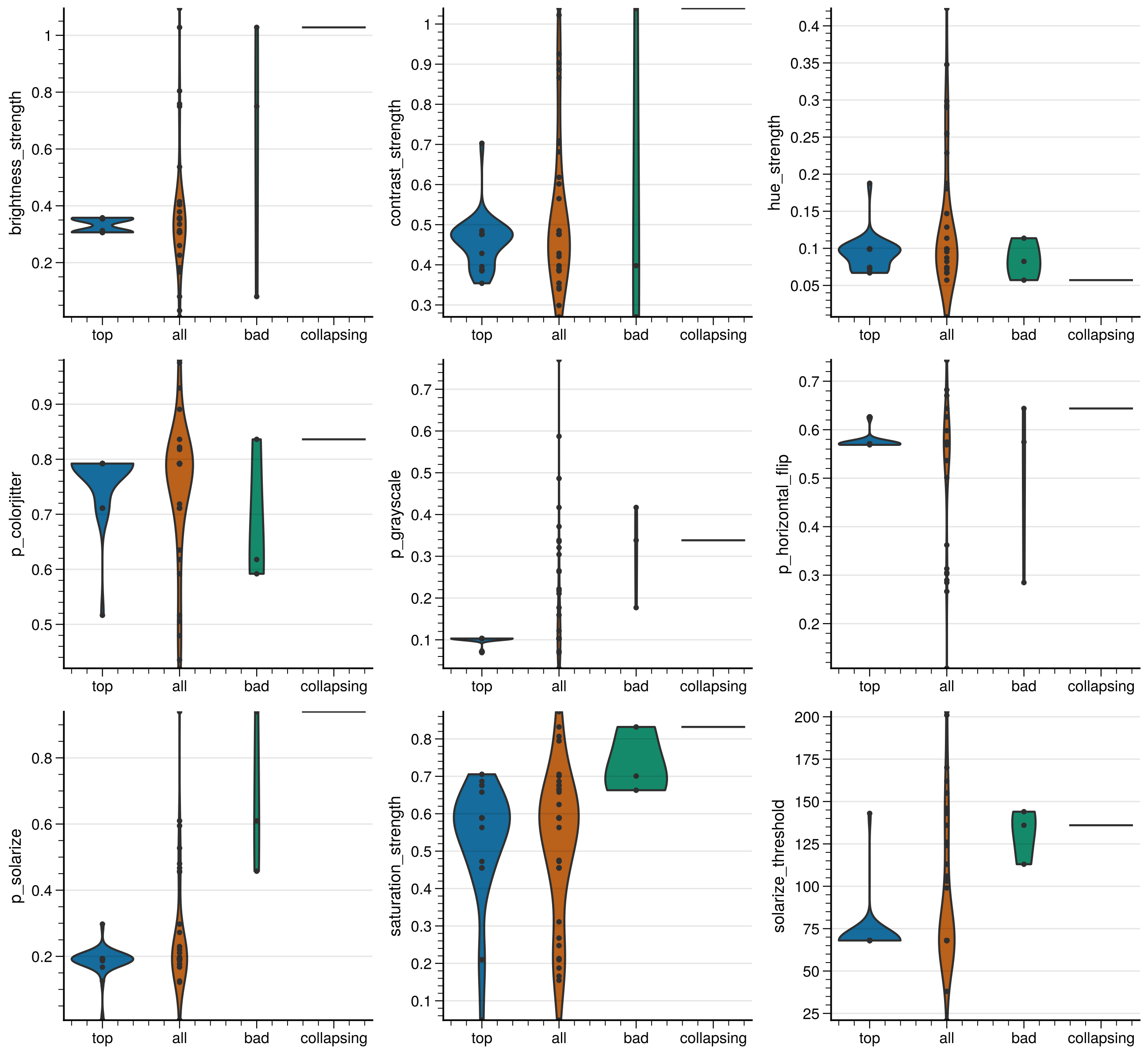

Figure 2. CIFAR-10 with SimSiam data augmentation search space.

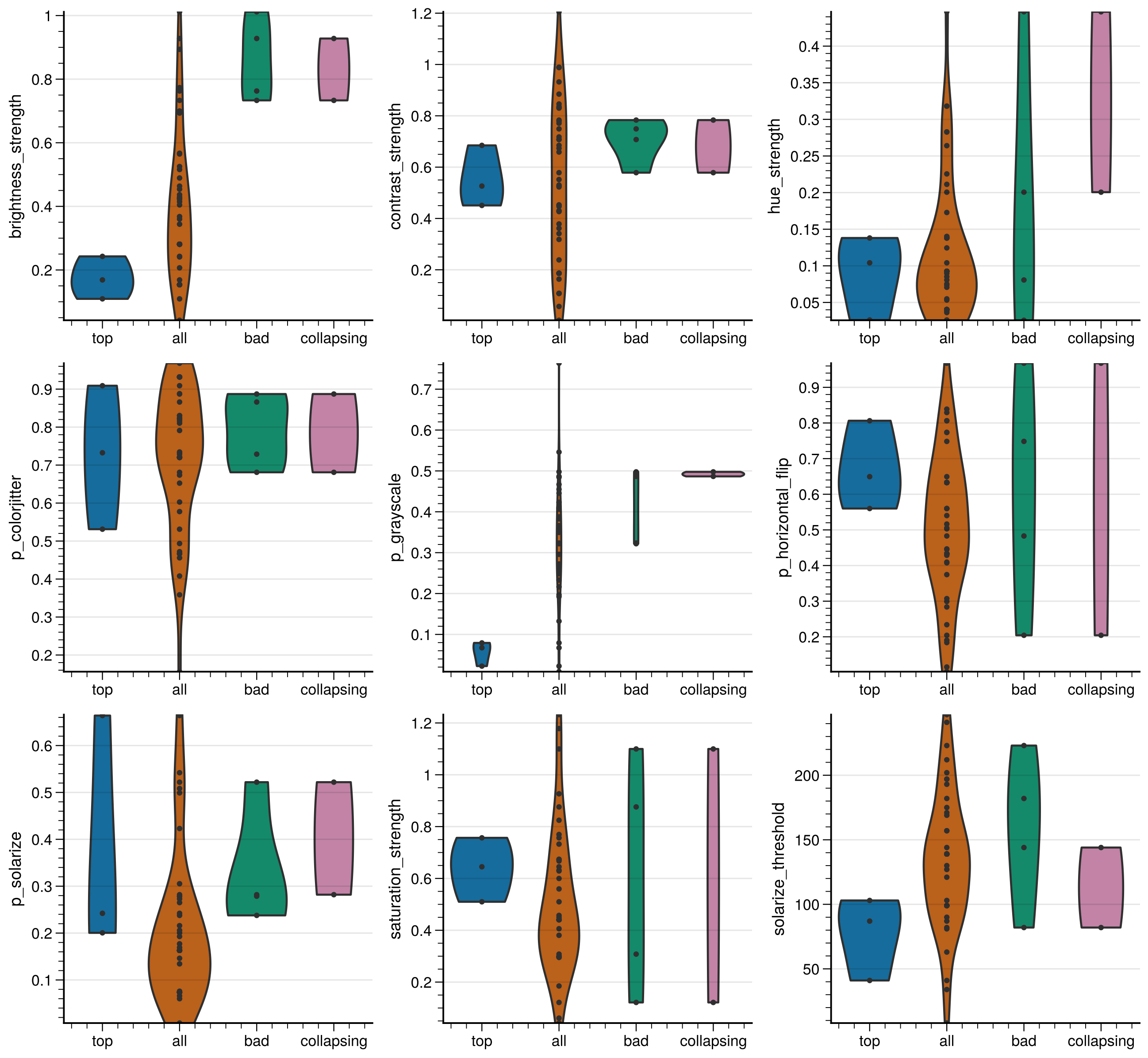

Figure 3. CIFAR-100 with SimSiam data augmentation search space.

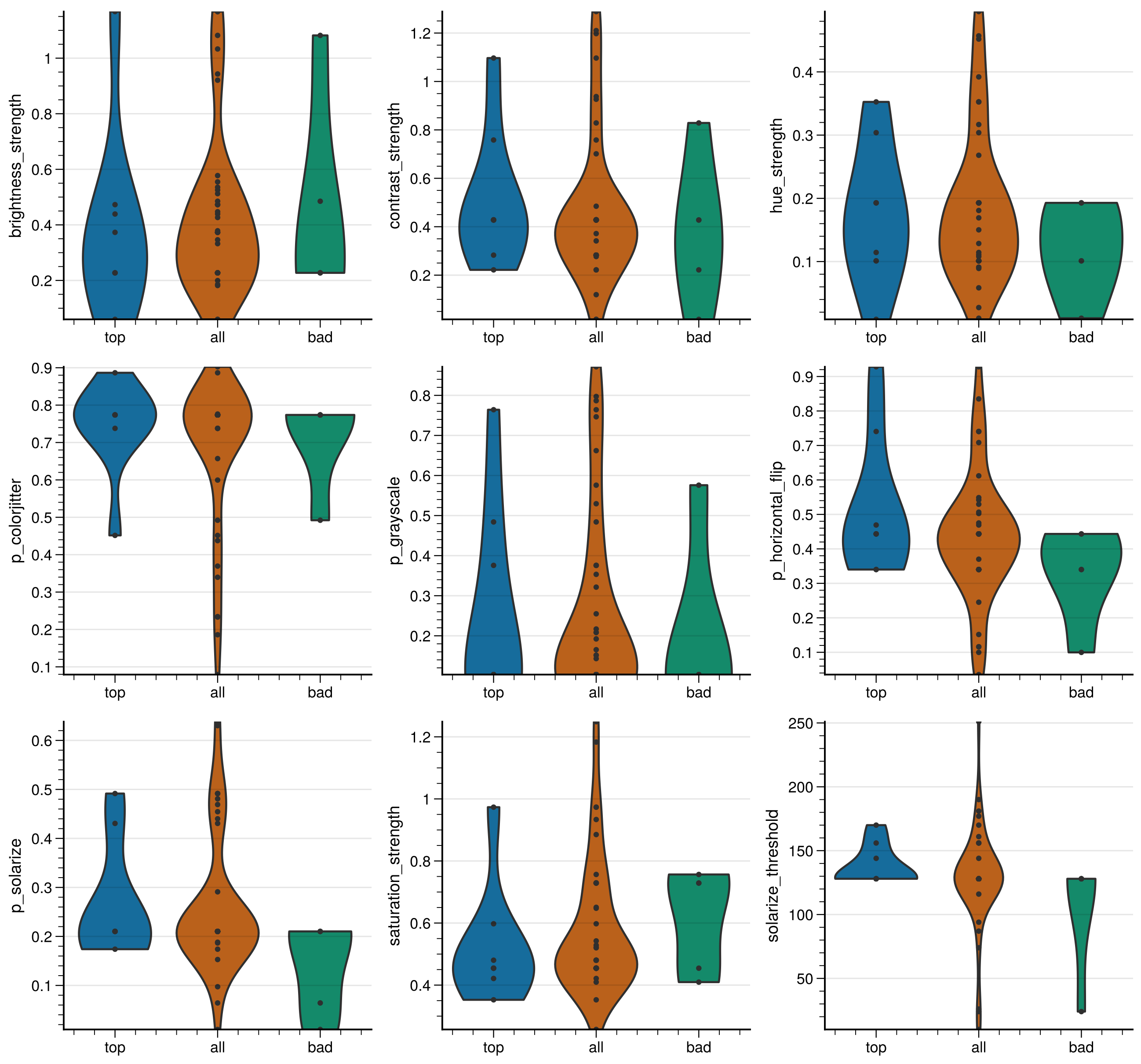

Figure 4. DermaMNIST with SimSiam data augmentation search space.

D.2. Tuned SimSiam Training Hyperparameters

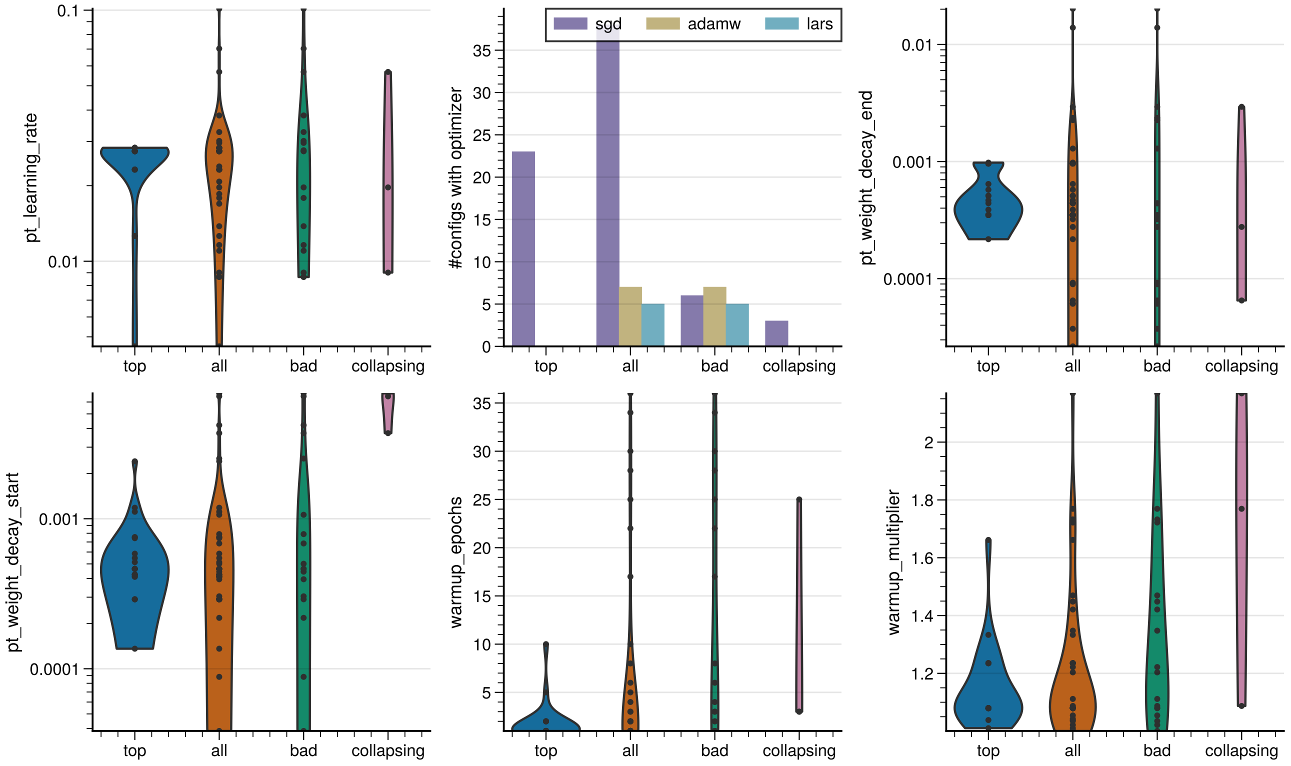

Figure 5. CIFAR-10 with SimSiam training hyperparameters search space.

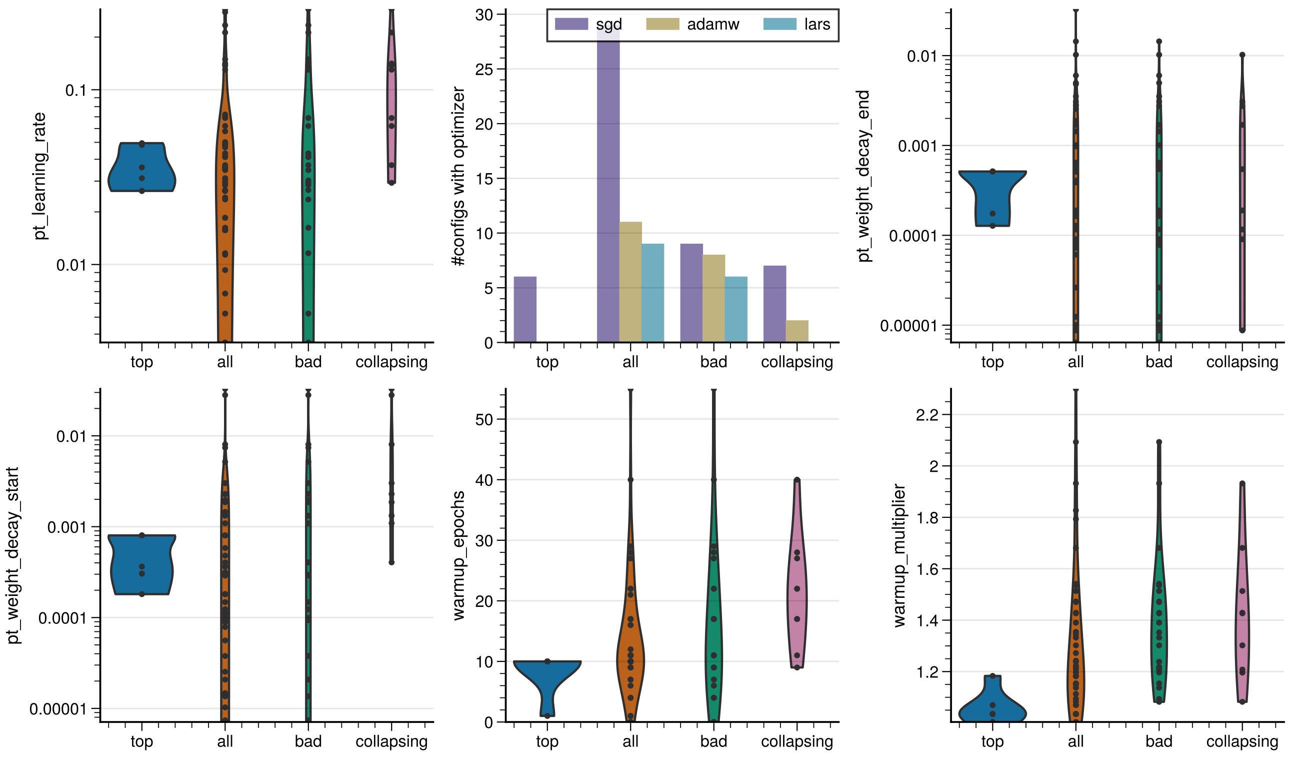

Figure 6. CIFAR-100 with SimSiam training hyperparameters search space.

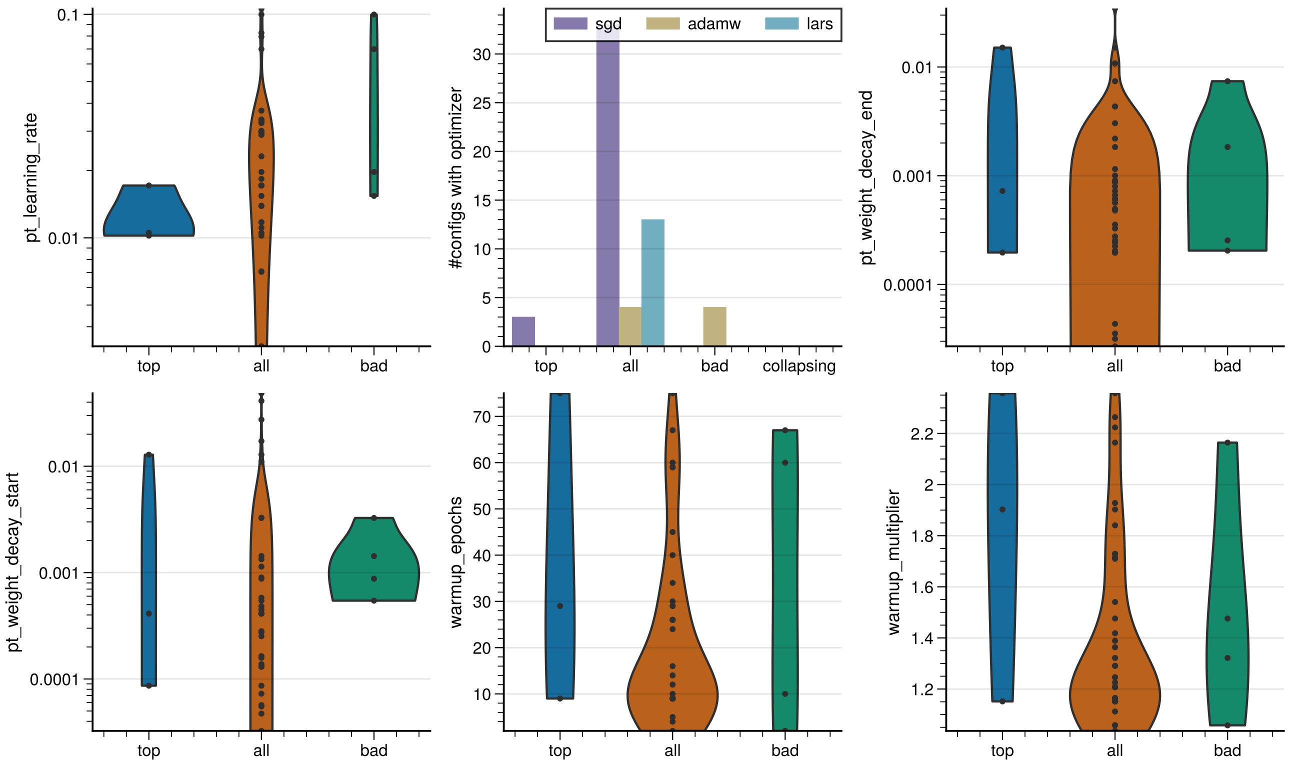

Figure 7. DermaMNIST with SimSiam training hyperparameters search space.