# Occam's Razor for Self Supervised Learning: What is Sufficient to Learn Good Representations?

**Mark Ibrahim**
FAIR, META
marksibrahim@meta.com

**David Klindt**
Cold Spring Harbor Laboratory
klindt@cshl.edu

**Randall Balestriero**
Brown University
Computer Science Department
rbalestr@brown.edu

## Abstract

Recent Self Supervised Learning (SSL) solutions have introduced numerous additional design choices, e.g., a projector network, positive views, or teacher-student networks. These additions limit the impact of theoretical studies that often fail to incorporate all those intertwined designs and slow-down the deployment of SSL methods to new domains as numerous hyper-parameters need to be carefully tuned. In this study, we demonstrate that for pretraining datasets of up to a few hundred thousands samples–the additional designs introduced by SSL can be removed without negatively impacting performances. That finding should tremendously simplify the practitioner's path to SSL deployment in numerous small and medium scale settings. In addition, our finding answers a long-lasting question: the often-experienced sensitivity to training settings and hyper-parameters encountered in SSL come from their design, rather than the absence of supervised guidance–as our simplified method is robust to changes in hyper-parameters and datasets.

## 1 Introduction

*Self-Supervised Learning* (SSL) [1, 2] has recently demonstrated that one can train, without labels, highly non-trivial Deep Neural Networks (DNNs) whose representations are often richer than supervised ones [3]. In particular, SSL differs from *reconstruction-based* methods such as (denoising,

SSL Workshop, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

**Algorithm 1** Our proposed simplified SSL solution: DIET

```
# take any preferred DNN e.g. resnet50
# see Algorithm 2 for other examples
f = torchvision.models.resnet50() # f_θ

# f comes with a classifier so we remove it
K = f.fc.in_features
f.fc = nn.Identity()

# define DIET's linear classifier and XEnt
W = nn.Linear(K, N, bias=False) # W in Eq. (1)
XEnt = nn.CrossEntropyLoss(label_smoothing=0.8)

# define dataset and train (Fig. 1)
train_dataset = DatasetWithIndices(train_dataset)
train_loader = DataLoader(train_dataset, ...)

for x, n in train_loader:
    loss = XEnt(W(f(x)), n) # Eq. (1)
    # backprop/optimizer/scheduler
```

```
from torch.utils.data import Dataset,
     DataLoader
from torchvision.datasets import
    CIFAR100

class DatasetWithIndices(Dataset):
    def __init__(self, dataset):
        self.dataset = dataset
    def __getitem__(self, n):
        # disregard the labels
        x, _ = self.dataset[n]
        return x, n
    def __len__(self):
        return len(self.dataset)

# example with CIFAR100
C100 = CIFAR100(root)
C100_w_ind = DatasetWithIndices(C100)
```

variational, masked) Autoencoders [4, 5, 6] and their variants by removing the need for a *decoder* DNN and an input-space reconstruction loss, both being difficult to design [7, 8, 9, 10]. Nonetheless, SSL which is the current state-of-the-art unsupervised learning solution, comes with many moving pieces, for instance, a carefully designed *projector* DNN $g_{\gamma}$ to perform SSL training with the composition $g_{\gamma} \circ f_{\boldsymbol{\theta}}$ and throwing away the projector ($g_{\gamma}$) afterwards [1], or advanced anti-collapse techniques involving moving average teacher models [11, 12], representation normalization [13, 14], or Entropy estimation [1, 15]. An incorrect pick of any of those moving pieces results in a drastic drop in performances [16, 17]. Most of those design choices have, however, been explored, carefully-tuned over many works, and set in stone when considering large scale natural images.

*But how can one deploy such pipelines to new label-free data modalities when so many design choices need to be carefully tuned?*

As of today, one would (i) either avoid learning altogether and use a pretrained model most commonly from Imagenet, or (ii) cross-validate again the many hyper-parameters of SSL models albeit requiring one to come up with some proxy to be used as validation metric. It is known that (i) can be highly sub-optimal when considering non natural images such as medical [18], and (ii) is practically unrealistic without labels as SSL losses fail to convey any qualitative information about the representation being learned [19, 20]. From the above, we argue that **Self Supervised Learning Needs Occam's Razor**–and we provide additional background and arguments in Appendix A.

To alleviate those limitations, we propose a stripped-down SSL method–coined DIET–that enables one to perform SSL on small to medium scale dataset without requiring labels. In fact, DIET is not sensitive to its (few) hyper-parameters and thus can be used as-is across data modalities, architectures, optimizers, . . . In addition, the DIET's loss value is also a strong indicator of quality of the representation. Hence, label-free evaluation is also made possible. We summarize below the key benefits of DIET:

1. **Competitive on common benchmarks and SOTA on medical and small datasets:** on more than 13 datasets including natural images (Table 1) and medical images (Section 3.2) even against SSL benchmarks pretrained on Imagenet (Table 5).
2. **Stable and Out-of-the-box:** consistently high performances without hyper-parameter tuning as tested across 16 architectures including ConvNexts, ViTs, 13 datasets, models, optimizers(Tables 1, 3 and 5).
3. **Practical, Efficient, and theory-friendly:** from the absence of positive pairs, projector networks, and decoders DIET training requires same resources as supervised learning. All our experiments are done on a single GPU. We also find DIET's training loss to strongly correlate with the downstream task test accuracy across architectures and datasets (Fig. 3).

An interactive demo is provided in this colab notebook.

## 2 DIET: Occam's Razor for Stable Small to Medium Scale SSL

The goal of this section is to introduce the proposed objective (Eq. (1)) that we will use to contrast with current SSL objectives. Thorough empirical validations on natural and medical images are provided in the next Sections 3.1 and 3.2

**Simplification 1: from relative to absolute loss.** Self Supervised Learning compares the inter-sample representations, aiming to *collapse* together the positive pairs, while ensuring that the entire representation does not collapse [21, 22, 23]. Within that formulation, SSL treats each sample and its views as a single class. We thus replace the relative inter-sample objective with a cross-entropy loss using as target the index of the original datum. That is, given a dataset of $N$ samples $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$, define the class of sample $\boldsymbol{x}_n$ with $n \in \{1, \dots, N\}$ to be $n$ itself.

**Simplification 2: removal of the nonlinear projector.** We remove the usual nonlinear projector ($g_{\gamma}$) and instead only use a liner classifier that maps the K-dimensional output of the originally considered model $f_{\boldsymbol{\theta}}$ to the $N$ classes of the cross-entropy objective. We denote that linear classifier as $\boldsymbol{W} \in \mathbb{R}^{N \times K}$,

**Simplification 3: removal of teacher-student network and positive pairs.** Additionally, we remove the need to have positive pairs and teacher-student network as the XEnt already prevents collapse and $\boldsymbol{W}$ "memorize" the class centroids, i.e., other positive views.

Table 1: **DIET often outperforms benchmarks on CIFAR100.** We employ the settings of Fig. 6, notice the consistent progression of the performance through architectures which is not easily achieved with standard SSL methods without per-architecture cross-validation. Benchmarks taken from † :[27]; ‡ :[28]; ∗:[29]; •:[30]; ⋄:[31]; ⋆:[32]; ◁:[33]; ▷:[34]; □:[35],1:[36], 2 :[37], IN100 provided in Table 4.

**CIFAR100**

| *Resnet18* | |
|---|---|
| SimCLR | $57.81^{\triangleright}$ |
| DINO | $58.12^{\bullet}$ |
| SimCO | $58.35^{*}$ |
| SimCLR+DCL | $58.50^{\dagger}$ |
| SimCLR | $60.30^{\ddagger}$ |
| SimCLR | $60.45^{\bullet}$ |
| W-MSE | $61.33^{\diamond}$ |
| SimCLR+CC | $61.91^{\ddagger}$ |
| BYOL | $62.01^{\bullet}$ |
| MoCoV2 | $62.34^{\bullet}$ |
| BYOL | $63.75^{\ddagger}$ |
| DIET | 63.77 |
| BYOL+CC | $64.62^{\ddagger}$ |
| SimSiam | $64.79^{\ddagger}$ |
| SwAV | $64.88^{\diamond}$ |
| SimCLR | $65.78^{\diamond}$ |
| SimSiam+CC | $65.82^{\ddagger}$ |

| *Resnet50* | |
|---|---|
| MoCoV2 | $53.44^{*}$ |
| SimMoCo | $54.64^{*}$ |
| SimCO | $58.48^{*}$ |
| SimCLR | $61.10^{\star}$ |
| SimCLR+DCL | $62.20^{\star}$ |
| MoCoV3 | $69.00^{\triangleleft}$ |
| DIET | 69.91 |
| *Resnet101* | |
| SimCLR | $52.28^{\dagger}$ |
| SimCLR+adv | $59.02^{\dagger}$ |
| MoCoV3 | $68.50^{\triangleleft}$ |
| DIET | 71.39 |
| *AlexNet* | |
| SplitBrain | $39.00^{\square}$ |
| InstDisc | $39.40^{\square}$ |
| DeepCluster | $41.90^{\square}$ |
| AND | $47.90^{\square}$ |
| DIET | 48.25 |
| SeLa | $57.40^{\square}$ |

**TinyImagenet**

| *Resnet18* | | *Resnet50* | |
|---|---|---|---|
| SimSiam | $44.54^{\ddagger}$ | SimCLR | $48.12^{2}$ |
| SimCLR | $46.21^{\ddagger}$ | SimSiam | $46.76^{2}$ |
| BYOL | $47.23^{\ddagger}$ | Spectral | $49.86^{2}$ |
| MoCo | $47.98^{\ddagger}$ | CorInfoMax | $54.86^{2}$ |
| SimCLR | $48.70^{1}$ | | |
| DINO | $49.20^{1}$ | | |
| DIET | | | |
| *resnet18* | 45.07 | *MLPMixer* | 39.32 |
| *resnet34* | 47.04 | *vit_b_16* | 48.38 |
| *wide_resnet50_2* | 50.03 | *densenet121* | 49.38 |
| *resnet50* | 51.66 | *convnext_small* | 50.05 |
| *resnet101* | 51.86 | *swin_t* | 50.80 |
| *resnext50_32x4d* | 52.45 | *convnext_tiny* | 50.88 |

Leading to the final formulation

$$\mathcal{L}_{\text{DIET}}(\boldsymbol{x}_n) = \text{XEnt}(\boldsymbol{W} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), n), \tag{1}$$

given a sample $\boldsymbol{x}_n \in \mathbb{R}^D$. We highlight that we couldn't find an existing study that proposed and experiment with Eq. (1). However, some close variations exist such as *Exemplar CNN* [24],*Instance Discrimination* [25] or *Noise as Targets* [26] which we discuss in Appendix C.

## 3 DIET: Stable Self Supervised Learning for any Modality and Model

To support the different claims we have made in the previous section, we will first explore natural image datasets in Section 3.1 and then move to medical images in Section 3.2 Throughout our study we rigorously follow the experimental setup described in Fig. 6 to demonstrate the stability of DIET, more specialized design choices should naturally lead to greater performance, if desired. Thorough ablation studies showing how DIET's performance are minimally impacted by optimizers, batch size, and other hyper-parameter changes are provided in Appendix D.

### 3.1 On Natural Images Against SSL With and Without Imagenet Pretraining

**DIET achieves high performance on CIFAR100:** Let's first consider CIFAR100 [38] with a few variations of Resnet [39] and AlexNet [40] architectures. On Alexnet, a few non-SSL baselines are available: SplitBrain [41], DeepCluster [42], InstDisc [25], AND [43], SeLa [44], and ReSSL [45]. Our models are trained with the DIET objective (Eq. (1)), and linear evaluation is employed to judge the quality of the learned representation. We report results Table 1 where we observe that DIET is able to match and often slightly exceed current SSL methods. Also, increasing the DNN capacity, *i.e.*, from Resnet18 to Resnet101 does not exhibit any overfitting using DIET.

**DIET achieves high performance on TinyImagenet and IN00:** We broaden the considered space of architectures to not only include the Resnet variants, but also SwinTransformers [46], VisionTransforms [47], Densenets [48], ConvNexts [49], WideResnets [50], ResNexts [51], and the MLPMixer [52]. We report those results in Table 1 where we observe that DIET is now around the average performance of the multiple SSL methods combined. As most SSL methods have been thoroughly tuned for Imagenet style tasks, we expect those benchmarks to be more challenging. That being said, the results from Table 1 demonstrate how DIET handles out-of-the-box any architecture change–even for different architecture families, e.g., with and without self-attention.

**DIET trained on small datasets competes with Imagenet pre-trained SSL.** We consider small datasets that are commonly handled by SSL through transfer learning: Aircraft [53], DTD [54], Pets [55], Flowers [56], CUB200 [57], Food101 [58], Cars [59]. By contrast, DIET finally provides an

| dataset | bloodmnist | dermamnist | pathmnist |
|---|---|---|---|
| DIET | 89.24 | 73.92 | **44.53** |
| DIET+ | **90.44** | **74.21** | **44.54** |
| MoCov2 | 53.70 | 66.88 | 18.97 |
| SimCLR | 14.56 | 66.88 | 11.80 |
| VICReg | 47.18 | 66.78 | 11.31 |
| Transfer | 88.13 | 74.06 | 59.37 |

Table 2: Performance on MedMNIST datasets using a Resnet18 (ViT provided in Table 8). DIET+ refers to the same DIET model trained for the same number of GPU hours as other models. VICReg is trained with the same hyperparameters as SimCLR with SGD 6e-2. Transfer is pretrained on ImageNet and fixed with a linear probe.

alternative approach by training directly on the such small dataset. We report those results in Table 5 where we see that DIET competes with or in some cases outperforms SSL models pretrained on much larger dataset.

We also find DIET can even outperform supervised learning methods in some cases when few data-labels are available. Furthermore, we show DIET's learning objective can be used with specialized network architectures such as scattering networks. We refer interested reader to Appendix J for details of these explorations.

### 3.2 On Medical Images

We now propose a more challenging comparison on medical images–an important modality that is often left behind when developing and tuning new SSL methods. We will see that DIET is able to produce state-of-the-art performances out-of-the-box.

We evaluate DIET training from scratch on three datasets from the MedMNISTv2 benchmark [60] (i) PathMNIST consisting of $90,000$ training images and $7,180$ test images, (ii) DermaMNIST consisting of $10,015$ training and $2,005$ test images, and finally (iii) BloodMNIST consisting of $17,092$ training and $3,421$ test images. To match a realistic unsupervised representation learning scenario, we employ for each method the hyper-parameters that work well on CIFAR100, and assume no labels are available for SSL training. For DIET, we use the same hyperparameters used for CIFAR100. For the baseline SSL methods, we select a variety of methods including a contrastive method (SimCLR), a momentum based method (MoCov2), and a recent non-contrastive method (ViCReg). For those, we use the default hyperparameters from [61] which yield good performance (> 80%) on CIFAR10, a comparable small dataset consisting of $60,000$ images.

We find that although all algorithms achieve high training accuracy via a linear probe as shown in Section 3.2, the features learned by the baseline SSL methods do not generalize well to the test sets. By contrast, DIET achieves much higher performances (also see Appendix K for DIET with ViT). We also show training curves for both the DIET loss and the online training accuracy which exhibit stable convergence out-of-the-box with the same hyper-parameters used throughout the paper in Fig. 10. In addition, DIET's simplicity makes it faster to reach a given number of epochs, specifically for ResNet18, DIET is 1.75x faster than SimCLR (and 1.72x faster than VICReg), thanks to DIET's simple learning objective.

## 4 Conclusions and Future Work

We examined current SSL pipelines and identified a few core components that clearly improve the quality of learned representations: (i) large number of training epochs, and (ii) strong and informed data augmentation. However, for numerous settings we explored, i.e., dataset with less than a few hundred thousands samples, the additional SSL complications, such as, positive views, nonlinear projector networks, teacher-student networks, do not help. On the contrary, we found that remove those additional parts of SSL pipelines make training much more stable and robust to changes in architecture, data modality, dataset size, and batch size. Even more surprising, the training objective now becomes informative of the downstream tasks test performance. We hope that our findings will help question which parts of our current pipelines are truly needed for case-by-case deployment, when knowing that they have been largely develop for large scale natural image tasks. Another impact of our findings lies the opening new doors to provable learning solutions. In fact, as the simpler pipeline we experimented with is easier to theoretically study, it could help in deriving novel and principled solutions.

## References


[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[3] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

[4] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[5] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[8] David B Grimes and Rajesh PN Rao. Bilinear sparse coding for invariant vision. *Neural computation*, 17(1):47–73, 2005.

[9] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[10] Romain Cosentino, Randall Balestriero, Yanis Bahroun, Anirvan Sengupta, Richard Baraniuk, and Behnaam Aazhang. Spatial transformer k-means. *arXiv preprint arXiv:2202.07829*, 2022.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[12] 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 IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[14] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[15] Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34:15543–15556, 2021.

[16] Romain Cosentino, Anirvan Sengupta, Salman Avestimehr, Mahdi Soltanolkotabi, Antonio Ortega, Ted Willke, and Mariano Tepper. Toward a geometrical understanding of self-supervised contrastive learning. *arXiv preprint arXiv:2205.06926*, 2022.

[17] Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Improving deep networks generalization by removing their head. *arXiv preprint arXiv:2206.13378*, 2022.

[18] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

[19] Arna Ghosh, Arnab Kumar Mondal, Kumar Krishna Agrawal, and Blake Richards. Investigating power laws in deep representation learning. *arXiv preprint arXiv:2202.05808*, 2022.

[20] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. *arXiv preprint arXiv:2210.02885*, 2022.

[21] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

[22] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022.

[23] Vivien Cabannes, Leon Bottou, Yann Lecun, and Randall Balestriero. Active self-supervised learning: A few low-cost relationships are all you need. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16274–16283, 2023.

[24] Dosovitskiy Alexey, Philipp Fischer, Jost Tobias, Martin Riedmiller Springenberg, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 99, 2015.

[25] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[26] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017.

[27] Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

[28] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16031–16040, 2022.

[29] Chaoning Zhang, Kang Zhang, Trung X Pham, Axi Niu, Zhinan Qiao, Chang D Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14441–14450, 2022.

[30] Trung Pham, Chaoning Zhang, Axi Niu, Kang Zhang, and Chang D Yoo. On the pros and cons of momentum encoder in self-supervised visual representation learning. *arXiv preprint arXiv:2208.05744*, 2022.

[31] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *J. Mach. Learn. Res.*, 23:56–1, 2022.

[32] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *European Conference on Computer Vision*, pages 668–684. Springer, 2022.

[33] Sucheng Ren, Huiyu Wang, Zhengqi Gao, Shengfeng He, Alan Yuille, Yuyin Zhou, and Cihang Xie. A simple data mixing prior for improving self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14595–14604, 2022.

[34] Kaiwen Yang, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. Identity-disentangled adversarial augmentation for self-supervised learning. In *International Conference on Machine Learning*, pages 25364–25381. PMLR, 2022.

[35] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: Bridging supervised and self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[36] Yann Dubois, Tatsunori Hashimoto, Stefano Ermon, and Percy Liang. Improving self-supervised learning by characterizing idealized representations. *arXiv preprint arXiv:2209.06235*, 2022.

[37] Serdar Ozsoy, Shadi Hamdan, Sercan Ö Arik, Deniz Yuret, and Alper T Erdogan. Self-supervised learning with an information maximization criterion. *arXiv preprint arXiv:2209.07999*, 2022.

[38] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[40] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[41] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1058–1067, 2017.

[42] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[43] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, pages 2849–2858. PMLR, 2019.

[44] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[45] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *Advances in Neural Information Processing Systems*, 34:2543–2555, 2021.

[46] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[47] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[48] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[49] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[52] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

[53] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[54] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[55] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[56] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[57] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Cub200 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[58] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[59] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[60] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1), jan 2023.

[61] Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, and Malte Ebner et al. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

[62] Deyu Meng, Yee Leung, Zongben Xu, Tung Fung, and Qingfu Zhang. Improving geodesic distance estimation based on locally linear assumption. *Pattern Recognition Letters*, 29(7):862–870, 2008.

[63] P Thomas Fletcher. Geodesic regression and the theory of least squares on riemannian manifolds. *International journal of computer vision*, 105(2):171–185, 2013.

[64] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[65] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

[66] Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[67] Matthew Brand and Kun Huang. A unifying theorem for spectral embedding and clustering. In *International Workshop on Artificial Intelligence and Statistics*, pages 41–48. PMLR, 2003.

[68] Yoshua Bengio, Jean-françcois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16, 2003.

[69] David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. *arXiv preprint arXiv:1806.02215*, 2018.

[70] Christian Lantuéjoul and Serge Beucher. On the use of the geodesic metric in image analysis. *Journal of Microscopy*, 121(1):39–49, 1981.

[71] Christian Lantuéjoul and Francis Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern recognition*, 17(2):177–187, 1984.

[72] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015.

[73] David L Donoho and Carrie Grimes. Image manifolds which are isometric to euclidean space. *Journal of mathematical imaging and vision*, 23(1):5–24, 2005.

[74] Michael B Wakin, David L Donoho, Hyeokho Choi, and Richard G Baraniuk. High-resolution navigation on non-differentiable image manifolds. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–1073. IEEE, 2005.

[75] Vivien Cabannes, Alberto Bietti, and Randall Balestriero. On minimal variations for unsupervised representation learning. *arXiv preprint arXiv:2211.03782*, 2022.

[76] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2021.

[77] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[78] Bobby He and Mete Ozay. Exploring the gap between collapsed & whitened features in self-supervised learning. In *International Conference on Machine Learning*, pages 8613–8634. PMLR, 2022.

[79] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019.

[80] Romain Cosentino, Sarath Shekkizhar, Mahdi Soltanolkotabi, Salman Avestimehr, and Antonio Ortega. The geometry of self-supervised learning models and its impact on transfer learning. *arXiv preprint arXiv:2209.08622*, 2022.

[81] Rachel Crowell. Why ai's diversity crisis matters, and how to tackle it. *Nature*, 2023.

[82] Christopher M Bishop. Mixture density networks. 1994.

[83] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[84] Yann LeCun. A path towards autonomous machine intelligence. *preprint posted on openreview*, 2022.

[85] Frank C Park. Distance metrics on the rigid-body motions with applications to mechanism design. 1995.

[86] Eero Simoncelli. A rotation invariant pattern signature. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 185–188. IEEE, 1996.

[87] James R Fienup. Invariant error metrics for image reconstruction. *Applied optics*, 36(32):8352–8357, 1997.

[88] Zhou Wang and Eero P Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii–573. IEEE, 2005.

[89] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[90] 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*, pages 16000–16009, 2022.

[91] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pages 3881–3890. PMLR, 2017.

[92] Wen Xu, Julian Jang-Jaccard, Amardeep Singh, Yuanyuan Wei, and Fariza Sabrina. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. *IEEE Access*, 9:140136–140146, 2021.

[93] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

[94] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[95] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[96] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[97] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021.

[98] Wenzheng Zhang and Karl Stratos. Understanding hard negatives in noise contrastive estimation. *arXiv preprint arXiv:2104.06245*, 2021.

[99] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[100] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[101] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[102] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, 30, 2017.

[103] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[104] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[105] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2208–2221, 2018.

[106] Shanel Gauthier, Benjamin Thérien, Laurent Alsene-Racicot, Muawiz Chaudhary, Irina Rish, Eugene Belilovsky, Michael Eickenberg, and Guy Wolf. Parametric scattering networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5749–5758, 2022.

# Supplementary Materials

The supplementary materials is providing the proofs of the main's paper formal results. We also provide as much background results and references as possible throughout to ensure that all the derivations are self-contained. Some of the below derivation do not belong to formal statements but are included to help the curious readers get additional insights into current SSL methods.

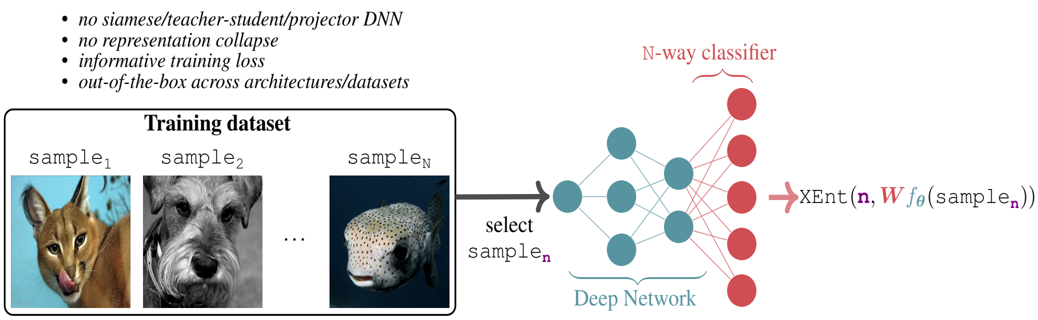


Figure 1: **DIET** uses the datum index (n) as the class-target –effectively turning unsupervised learning into a supervised learning problem. In our case, we employ the cross-entropy loss (X-Ent), no extra care needed to handle different dataset or architectures. As opposed to current SOTA, we do not rely on a projector nor positive views *i.e* no change needs to be done to any existing supervised pipeline to obtain DIET. As highlighted in Fig. 3, DIET's training loss is even informative of downstream test performances, and as ablated in Appendix D there is no degradation of performance with longer training, even for very small datasets (Table 5).

## A Why Self Supervised Learning Needs Occam's Razor

Unsupervised learning often takes the form of intricate methods combining numerous moving pieces that need readjustment for each DNN architecture and dataset. As a result reproducibility, transferability across domains, and explainability are hindered.

**Spectral embedding is computationally challenging.** Spectral embedding takes many forms but can be summarized into estimating geodesic distances [62, 63] between all or some pairs of training samples to then learn a non-parametric [64, 65, 66, 67], or parametric [68, 69] mapping that produces embeddings whose pairwise distances matches the estimated geodesic ones. As such, spectral embedding heavily relies on the estimation of the geodesic distances which is a challenging problem [70, 71, 72], especially for images and videos [73, 74]. This limitation motivated the development of alternative methods, *e.g.*, Self-Supervised Learning (SSL) that often employ losses similar to spectral embedding [21, 22, 75] but manage to move away from geodesic distance estimation through the explicit generation of positive pairs, *i.e.*, that are close neighbors on the data manifold.

**Self-Supervised Learning is over-specialized.** Despite impressive performance and rigorous theoretical motivation, SSL development was mostly driven by industry driven research and thus entirely focused on large-scale natural images and sounds. In fact, SSL has evolved to a point where novel methods are architecture and dataset specific. A few challenges that limit SSL to be widely adopted are (i) loss values which are uninformative of the DNN's quality [20, 76], partly explained by the fact that SSL composes the DNN of interest $f_\theta$ with a projector DNN $g_\gamma$ appended to it during training and discarded afterwards, (ii) too many per-loss and per-projector hyper-parameters whose impact on the DNN's performances are hard to control or predict [11, 77, 78], and (iii) lack of transferability of the hyper-parameters across datasets and architectures [79, 80]Lastly, SSL requires heavy code refactoring, e.g., it requires to generate positive pairs and forward them to siamese DNNs, sometimes with one DNN having parameters as the moving average of the other. This makes SSL implementation more costly than supervised learning often requiring distributed training and long training schedules that, effectively, reduce the accessibility and inclusivity of SSL research [81].

**Reconstruction-based learning is unstable.** Reconstruction without careful tuning of the loss has been known to be sub-optimal for long [82, 83] and new studies keep reminding us of that [84]. The argument is simple, suppose one aims to minimize a reconstruction metric $R$ for some input $\boldsymbol{x}$

$$R(d_\gamma(e_\eta(\boldsymbol{x})), \boldsymbol{x}), \tag{2}$$

where $e_\eta$ and $d_\gamma$ are parametrized learnable encoder and decoder networks respectively; $e_\eta(\boldsymbol{x})$ is the representation of interest to be used after training. In practice, as soon as some noise $\epsilon$ is present in the data, *i.e.* we observe $\boldsymbol{x} + \epsilon$ and not $\boldsymbol{x}$, that noise $\epsilon$ must be encoded by $e_\eta$ to minimize the loss from Eq. (2) unless one carefully designs $R$ so that $R(\boldsymbol{x} + \epsilon, \boldsymbol{x}) = 0$. However, designing such a *noise invariant* $R$ has been attempted for decades [8, 85, 86, 87, 88] and remains a challenging open problem. Hence, many solutions rely on learning $R$, e.g., in VAE-GANs [9] bringing even further instabilities and training challenges. Other alternatives carefully tweak $R$ per dataset and architectures, e.g., to only compute the reconstruction loss on parts of the data as with BERT [89] or MAEs [90]. Lastly, the quality of the encoder representation depends on its architecture but also on the decoder [91, 92] making cross-validation more costly and unstable [93].

SSL is the family of method that have produced the most significant state-of-the-art solutions in recent years. Hence, it is the solution of choice that any practitioner hopes to deploy. As such, we propose to take a step towards understanding and alleviating the many practical challenges that would be up against through **DIET**–a stripped down SSL pipeline.

# B Benefits for Practical Deployment and Theoretical Research

There are many direct benefit of the DIET's objective emerging from its simplicity. We highlight both a theoretical and a practical benefit.

**Benefit for theoretical research and provable guarantees.** First, DIET opens numerous avenues for theoretical research. This is in sharp contrast with the original SSL methods. In fact, current SSL lacks of theoretical guarantees as all existing studies have derived optimality conditions at the projector's output [21, 36, 94, 95, 96, 97, 98, 99] which is not the output of interest since the projector is thrown away after SSL training and the DNN's output and the projector's output greatly differ [1, 16, 17, 100]. As a further demonstration of DIET's theory-friendliness, we propose in Appendix E a theoretical study of DIET with a linear model $f_{\boldsymbol{\theta}}$, in which case we are able to prove that DIET performs a low-rank decomposition of the input data matrix and provably recovers the data's principal components. Again, that last result highlights how Eq. (1) greatly reduces the barrier to derive novel theoretical resutls and guarantee for SSL.

**Benefits for practical development and deployment** Second, the amount of code refactoring is minimal: there is no change required for the data loading pipelines as opposed to SSL which requires positive pairs, no need to specify teacher-student architectures, and no need to design a projector/predictor DNN. Second, DIET's implementation is not architecture specific as we validate on Resne(x)ts, ConvNe(x)ts, Vision Transformers and their variants. Furthermore, DIET does not introduce any additional hyper-parameters in addition to the ones already present in supervised learning–and because DIET's training loss is informative of test classification performances (Fig. 3)–it opens the door to truely label-free SSL.

# C Relation Between DIET and Existing SSL

Despite DIET's simplicity, we could not find an existing method that considered it perhaps due to the common belief that dealing with hundreds of thousands of classes ($N$ in Fig. 1, the training set size) would not produce successful training. As such, the closest method to ours is *Exemplar CNN* [24] which extracts a few patches from a given image dataset, and treats each of them as their own class; this way the number of classes is the number of extracted patches, which is made independent from $N$. A more recent method, *Instance Discrimination* [25] extends this by introducing inter-sample discrimination. However, they do so using a non-parametric softmax, *i.e.*, by defining a learnable bank of centroids to cluster training samples; for successful training those centroids must be regularized to prevent representation collapse. As we will compare in Table 1, DIET outperforms Instance Discrimination and Exemplar CNN while being simpler. Lastly, methods such as *Noise as Targets* [26] and DeepCluster [42] are quite far from DIET as (i) they perform clustering and use the datum's cluster as its class, *i.e.*, greatly reducing the dependency on $N$; and (ii) they perform clustering in the output space of the model $f_{\boldsymbol{\theta}}$ being learned which brings multiple collapsed solutions that force those

Table 3: Ablation studies indicate that **DIET benefits from longer training and stronger data augmentation while being robust to architecture and batch-size changes**. We report top1 test accuracy on CIFAR100 with varying training epochs (**top left**), on TinyImagenet with varying DA pipelines (Algorithm 3), and on TinyImagenet with 3k training epochs and with varying batch-size (**bottom**) with learning rate $0.001\frac{\text{bs}}{256}$; additional comparisons on MedMNIST Table 6.

| Epochs | 50 | 100 | 200 | 500 | 1000 | 5000 | 10000 |
|---|---|---|---|---|---|---|---|
| resnet18 | 33.46 | 42.94 | 48.24 | 54.54 | 58.81 | 62.63 | 63.29 |
| resnet50 | 37.71 | 47.86 | 54.04 | 60.23 | 64.24 | 69.51 | 69.91 |
| resnet101 | 34.03 | 46.59 | 54.3 | 60.8 | 64.71 | 70.56 | 71.39 |

| DA strength | 1 | 2 | 3 |
|---|---|---|---|
| resnet18 | 31.48 | 43.62 | 43.88 |
| resnet34 | 32.93 | 45.60 | 45.75 |
| resnet50 | 40.24 | 48.80 | 50.81 |
| resnet101 | 40.07 | 49.74 | 50.76 |

| batch-size | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|---|---|---|---|
| resnet18 | 32.9 | 37.9 | 42.7 | 43.4 | 43.3 | 43.7 | 43.7 | 42.6 |

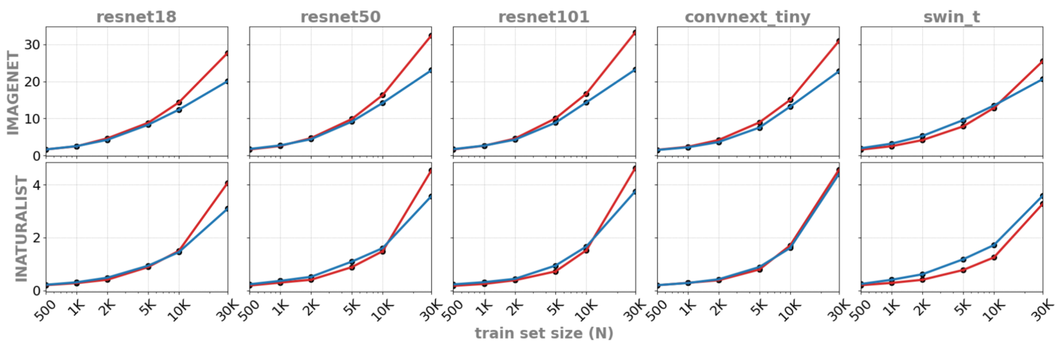


Figure 2: **DIET matches supervised learning on datasets with only a few samples per class.** Depiction of DIET's downstream performances (blue) against supervised learning (red) controlling training set size (**x-axis**); evaluation is performed over the original full evaluation set. DIET is able to learn highly competitive representations when the dataset is small with only a few samples per classes. See Fig. 7 for additional datasets.

methods to employ complicated mechanisms to ensure training to learn non-trivial representations. We note that while the added complexity enables those methods to scale to large datasets, it also greatly increases the performance sensitivity to the training hyper-parameters. We also emphasize in details why the simplifications put into DIET are crucial not only for deployment and theoretical guarantees in Appendices B and E.

# D DIET's Dependency on Data-Augmentation, Training Time and Batch Size

The aim of this section is to better inform practitioners about the role of Data-Augmentations (DA), training time, and label smoothing in DIET's performances; as well as sensitivity to batch size, which is crucial for single-GPU training.

**Batch-size does not impact DIET's performance.** One important question when it comes to training a method with low resources is the ability to employ (very) small batch sizes. This is in fact one reason hindering the deployment of SSL methods which require quite large batch sizes to work (256 is a strict minimum in most cases). Therefore, we perform a small sensitivity analysis in Table 3 where we vary the batch size from 8 to 2048 without any hyper-parameter tuning other than the standard learning rate scaling used in supervised learning: $lr = 0.001\frac{\text{bs}}{256}$. We observe small fluctuations of performances (due to a sub-optimal learning rate) but no significant drop in performance, even for batch size of 32. When going to 16 and 8, we observe slightly lower performances, likely due to batch-normalization [101] which is known to behave erratically below a batch size of 32 [102].
**Data-Augmentation sensitivity is similar to SSL.** We observed in the previous Section 3.1 that when using DA, DIET is able to perform on par with highly engineered state-of-the-art methods. Yet, knowing which DA to employ is not trivial, e.g., many data modalities have no obvious DA. One natural question is, thus, concerning the sensitivity of DIET's performance to the employed DA. To that end, we propose three DA regimes, one only consistent of random crops and horizontal flips (**strength:1**), which could be considered minimal in computer vision, one which adds color jittering and random grayscale (**strength:2**), and one last which further adds Gaussian blur and random erasing

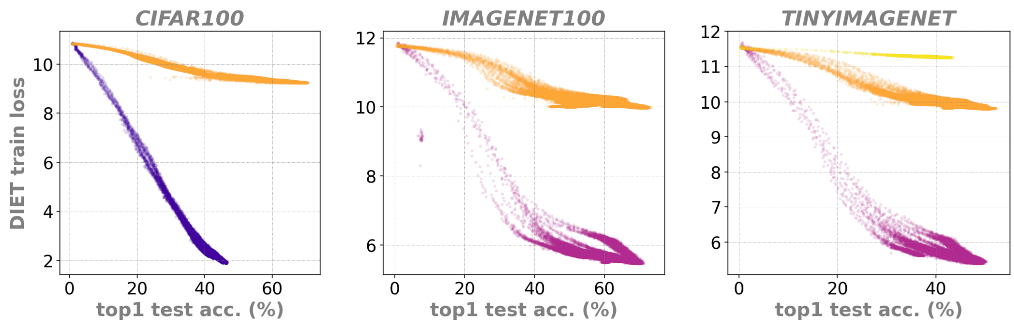


Figure 3: **DIET's training loss is indicative of downstream test performance.** We depict DIET's training loss (**y-axis**) against the online test linear probe accuracy (**x-axis**) for all the models, hyper-parameters, and training epochs. Yellow to purple correspond to different label smoothing which plays a role in DIET's convergence speed (Appendix D). For a given label smoothing parameter, there exists a strong relationship between **DIET**'s training loss and the downstream test accuracy enabling label-free quantitative quality assessment one's model.

Table 4: **DIET is competitive and works out-of-the-box across architectures.** We keep the settings of Fig. 6, as per Table 1. Benchmarks from 1:[36], 2 :[37].

**Imagenet-100 (IN100)**

| *Resnet18* | |
|---|---|
| SimMoCo | 58.20$^{*}$ |
| MocoV2 | 60.52$^{*}$ |
| SimCo | 61.28 $^{*}$ |
| W-MSE2 | 69.06 $^{2}$ |
| ReSSL | 74.02$^{\bullet}$ |
| DINO | 74.16$^{\bullet}$ |
| MoCoV2 | 76.48$^{\bullet}$ |
| BYOL | 76.60$^{\bullet}$ |
| SimCLR | 77.04$^{2}$ |
| SimCLR | 78.72$^{2}$ |
| MocoV2 | 79.28$^{2}$ |
| VICReg | 79.40$^{2}$ |
| BarlowTwins | 80.38$^{2}$ |

| *Resnet50* | |
|---|---|
| MoCo+Hyper. | 75.60 $^{\star}$ |
| MoCo+DCL | 76.80 $^{\star}$ |
| MoCoV2 + Hyper. | 77.70 $^{\star}$ |
| BYOL | 78.76 $^{2}$ |
| MoCoV2 + DCL | 80.50 $^{\star}$ |
| SimCLR | 80.70 $^{\star}$ |
| SimSiam | 81.60$^{2}$ |
| SimCLR + DCL | 83.10 $^{\star}$ |

| DIET | | | |
|---|---|---|---|
| *resnet18* | 64.31 | *resnet50* | 73.50 |
| *wide_resnet50_2* | 71.92 | *convnext_small* | 71.06 |
| *resnext50_32x4d* | 73.07 | *MLPMixer* | 56.46 |
| *densenet121* | 67.46 | *swin_t* | 67.02 |
| *convnext_tiny* | 69.77 | *vit_b_16* | 62.63 |

[103] (**strength:3**); the exact parameters for those transformations are given in Algorithm 3. We observe on TinyImagenet and with a Resnet34 the following performances 32.93$\pm$ 0.6, 45.60$\pm$ 0.2, and 45.75$\pm$ 0.1 respectively over 5 independent runs, details and additional architectures provided in Fig. 9 and Table 3 in the Appendix. We thus observe that while DIET greatly benefit from richer DA (strength:1 $\mapsto$ 2), it however does not require heavier transformation such as random erasing.

**Label smoothing helps.** One important difference in training behavior between supervised learning and SSL is in the number of epochs required to see the quality of the representation plateau. Due to the different loss used in DIET, one might wonder about the differences in training behavior. We observe that DIET takes more epochs than SSL until the loss converges. However, by using large values of label smoothing, *e.g.*, 0.8, it is possible to obtain faster convergence. We provide a sensitivity analysis in Fig. 8 and Table 3 in the Appendix. In fact, one should recall that within a single epoch, only one of each datum/class is observed, making the convergence speed of the classifier's $\boldsymbol{W}$ matrix the main limitation; we aim to explore improved training strategies in the future as discussed in Section 4.

Table 5: **DIET trained on small datasets competes with Imagenet pre-trained SSL.** We also report performances for a ViT based architecture (SwinTiny) to demonstrate the ability of DIET to handle different models out-of-the-box following Fig. 6. Benchmarks from †:[34], +:[104]

| Arch. | Pretrain | Frozen | N= / C= | *Aircraft* 6667 / 100 | *DTD* 1880 / 47 | *Pets* 2940 / 37 | *Flower* 1020 / 102 | *CUB-200* 11788 / 200 | *Food101* 68175 / 101 | *Cars* 6509 / 196 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Resnet18* | IN100† | Yes | SimCLR | 24.19 | 54.35 | 46.46 | 75.00 | 16.73 | - | - |
| | | | +CLAE | 25.87 | 52.12 | 43.55 | 76.82 | 17.58 | - | - |
| | | | +IDAA | 26.02 | 54.97 | 46.76 | 77.99 | 18.15 | - | - |
| | None | No | DIET | 37.29 | 50.62 | 64.06 | 72.01 | 33.03 | 62.00 | 42.55 |
| *Resnet50* | IN-1k+ | Yes | InsDis | 36.87 | 68.46 | 68.78 | 83.44 | - | 63.39 | 28.98 |
| | | | MoCo | 35.55 | 68.83 | 69.84 | 82.10 | - | 62.10 | 27.99 |
| | | | PCL. | 21.61 | 62.87 | 75.34 | 64.73 | - | 48.02 | 12.93 |
| | | | PIRL | 37.08 | 68.99 | 71.36 | 83.60 | - | 64.65 | 28.72 |
| | | | PCLv2 | 37.03 | 70.59 | 82.79 | 85.34 | - | 64.88 | 30.51 |
| | | | SimCLR | 44.90 | 74.20 | 83.33 | 90.87 | - | 67.47 | 43.73 |
| | | | MoCov2 | 41.79 | 73.88 | 83.30 | 90.07 | - | 68.95 | 39.31 |
| | | | SimCLRv2 | 46.38 | 76.38 | 84.72 | 92.90 | - | 73.08 | 50.37 |
| | | | SeLav2 | 37.29 | 74.15 | 83.22 | 90.22 | - | 71.08 | 36.86 |
| | | | InfoMin | 38.58 | 74.73 | 86.24 | 87.18 | - | 69.53 | 41.01 |
| | | | BYOL | 53.87 | 76.91 | 89.10 | 94.50 | - | 73.01 | 56.40 |
| | | | DeepClusterv2 | 54.49 | 78.62 | 89.36 | 94.72 | - | 77.94 | 58.60 |
| | | | Swav | 54.04 | 77.02 | 87.60 | 94.62 | - | 76.62 | 54.06 |
| | None | No | DIET | 44.81 | 51.75 | 67.08 | 73.32 | 41.03 | 71.58 | 55.82 |
| *SwinTiny* | None | No | DIET | 33.15 | 51.88 | 58.06 | 70.78 | 32.11 | 68.86 | 47.12 |
| *Convnext-S* | None | No | DIET | 43.13 | 49.52 | 61.72 | 67.72 | 31.44 | 69.84 | 40.63 |

# E Linear Model Analysis

Let's consider the case of a linear model followed by the DIET loss. So the modeling loss given the data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, the linear mapping matrix $\boldsymbol{V} \in \mathbb{R}^{D \times K}$ and the DIET linear probe matrix $\boldsymbol{W} \in \mathbb{R}^{N \times K}$, is of the form

$$\begin{aligned}\mathcal{L} =& \text{CrossEntropy}(\boldsymbol{I}, \boldsymbol{X}\boldsymbol{V}\boldsymbol{W}^\top)\\ =&\sum_{n=1}^{N} -(\boldsymbol{X}\boldsymbol{V}\boldsymbol{W}^\top)_{n,n} + \log\left(\sum_{m=1}^{N}\exp((\boldsymbol{X}\boldsymbol{V}\boldsymbol{W}^\top)_{n,m})\right)\\ =&\sum_{n=1}^{N} -\langle(\boldsymbol{X}\boldsymbol{V})_{n,.}, (\boldsymbol{W})_{n,.}\rangle + \log\left(\sum_{m=1}^{N}\exp\left(\langle(\boldsymbol{X}\boldsymbol{V})_{n,.}, (\boldsymbol{W})_{m,.}\rangle\right)\right),\end{aligned}$$

the derivative with respect to the parameters $\boldsymbol{V}$ and $\boldsymbol{W}$ are given by

$$\nabla_{\boldsymbol{W}} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{V},\ \nabla_{\boldsymbol{V}} = \boldsymbol{X}^\top \boldsymbol{A}\boldsymbol{W}$$

where the matrix $\boldsymbol{A}$ is given by

$$(\boldsymbol{A})_{i,j} = \left(\frac{e^{(\boldsymbol{X}\boldsymbol{V}\boldsymbol{W}^\top)_{i,j}}}{\sum_{n=1}^{N} e^{(\boldsymbol{X}\boldsymbol{V}\boldsymbol{W}^\top)_{i,n}}} - 1_{\{i=j\}}\right).$$

The above analysis is true for any matrix $\boldsymbol{X}, \boldsymbol{V}, \boldsymbol{W}$. Finding a general solution by setting the gradient to 0 is not trivial due to the $\boldsymbol{A}$ matrix involving a softmax operation. However, for a special class of data matrices $\boldsymbol{X}$, we are able to find a closed-form optimal solution for $\boldsymbol{V}$ and $\boldsymbol{W}$. Let's now consider the following low-rank model for the input data matrix $\boldsymbol{X}$ as

$$\boldsymbol{X} \triangleq [\mu_1, \dots, \mu_1, \dots, \mu_K, \dots, \mu_K]^\top,$$

where each $\mu_i$ is repeated $N/K$ times. That is, we assume that $\boldsymbol{X}$ has a low-rank structured made of "centroids". Note that while we assume here that each centroid is repeated the same number of times to simplify notations, none of the following results require uniform distribution of the centroids.

Then, DIET will effectively learn the clustering, as per the SVD of the data. In fact, let's consider the following parameters $\boldsymbol{V} = \boldsymbol{V}_{\boldsymbol{X}}\boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}$ and $\boldsymbol{W} = \kappa \boldsymbol{U}_{\boldsymbol{X}}$ where we used the (reduced) singular value

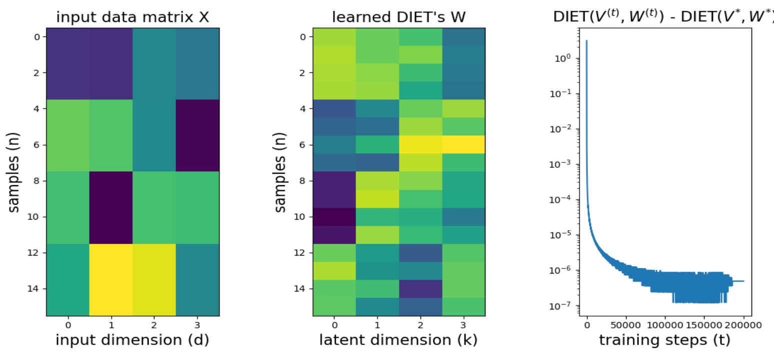


Figure 4: Empirical validation of Appendix E depicting the optimal solution for DIET for the parameters $\boldsymbol{W}$ and $\boldsymbol{V}$ under a clustered input data assumption (**left column**), in this case, made of four clusters with four samples per cluster. The learned $\boldsymbol{W}$ given in the **middle column** converge to the same clustering, as predicted by our closed-form solution. We also obtain in the **right column** the evolution of the DIET training loss that we compare against the optimal value of the loss (obtained from the optimal parameters). We see that the training converges towards the optimal value of the loss (up to 1e-7 at the end of that training episode).

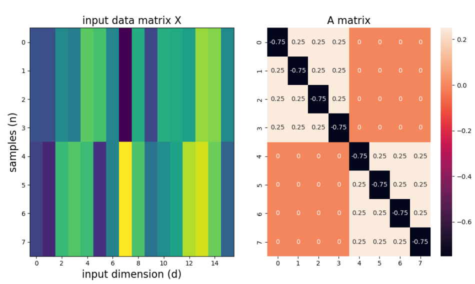


Figure 5: Depiction of the optimal $\boldsymbol{A}$ matrix (recall Eq. (3)) on the **right**, obtained empirically from inserting the optimal parameters that we found for $\boldsymbol{W}$ and $\boldsymbol{V}$. As predicted by Eq. (3) that matrix is made of blocks aligned with the original clustering of the input data matrix $\boldsymbol{X}$ given on the **left**.

decomposition of $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}_{\boldsymbol{X}} \boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{V}_{\boldsymbol{X}}^{\top}$, and with $\kappa \gg 0$. In that setting, the matrix $\boldsymbol{A}$ becomes with a block structure as per

$$(\boldsymbol{A})_{i,j} = \frac{1}{K} 1_{\{[i/(N/K)]=[j/(N/K)]\}} - 1_{\{i=j\}}, \tag{3}$$

as depicted in Fig. 5. This leads to a zero-gradient

$$\nabla_{\boldsymbol{W}} = \mathbf{0}, \nabla_{\boldsymbol{V}} = \mathbf{0},$$

effectively showing that we obtain the optimal parameters, as depicted in Fig. 4.

**DIET's experimental setup:**

- Official Torchvision architectures (no changes in init./arch.), only swapping the classification layer with DIET's one (right of Fig. 1), no projector DNN
- Same DA pipeline ($\mathcal{T}$ in Fig. 1) across datasets/architectures with batch size of 256 to fit on 1 GPU
- AdamW optimizer with linear warmup (10 epochs) and cosine annealing learning rate schedule, `XEnt` loss (right of Fig. 1) with *label smoothing of* 0.8
- *Learning rate/weight-decay* of 0.001/0.05 for non transformer architectures and 0.0002/0.01 for transformers

Figure 6: In underlined are the design choices directly ported from standard supervised learning (not cross-validated for DIET), in *italic* are the design choices cross-validated for DIET but held constant across this study unless specified otherwise. Batch-size sensitivity analysis is reported in Table 3 and Fig. 9 showing that performances do not vary when taking values from 32 to 4096. `XEnt`'s label smoothing parameter plays a role into DIET's convergence speed, and is cross-validated in Fig. 8 and Table 3; we also report DA ablation in Fig. 9 and Table 3.

# F Code

**Algorithm 2** Get the output dimension and remove the linear classifier from a given torchvision model (Pytorch used for illustration).

```
model = torchvision.models.__dict__[architecture]()

# CIFAR procedure to adjust to the lower image resolution
if is_cifar and "resnet" in architecture:
    model.conv1 = torch.nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=2, bias=False)
    model.maxpool = torch.nn.Identity()

# for each architecture, remove the classifier and get the output dim. (K)
if "alexnet" in architecture:
    K = model.classifier[6].in_features
    model.classifier[6] = torch.nn.Identity()
elif "convnext" in architecture:
    K = model.classifier[2].in_features
    model.classifier[2] = torch.nn.Identity()
elif "convnext" in architecture:
    K = model.classifier[2].in_features
    model.classifier[2] = torch.nn.Identity()
elif "resnet" in architecture or "resnext" in architecture or "regnet" in architecture:
    K = model.fc.in_features
    model.fc = torch.nn.Identity()
elif "densenet" in architecture:
    K = model.classifier.in_features
    model.classifier = torch.nn.Identity()
elif "mobile" in architecture:
    K = model.classifier[-1].in_features
    model.classifier[-1] = torch.nn.Identity()
elif "vit" in architecture:
    K = model.heads.head.in_features
    model.heads.head = torch.nn.Identity()
elif "swin" in architecture:
    K = model.head.in_features
    model.head = torch.nn.Identity()
```

## F.1 Pushing the DIET to Large Models and Datasets

Given DIET's formulation of considering each datum as its own class, it is natural to ask ourselves how scalable is such a method. Although we saw that on small and medium scale dataset, DIET's was able to come on-par with most current SSL methods, it is not clear if this remains true for larger datasets. In this section we briefly describe what can be done to employ DIET on datasets such as Imagenet and INaturalist.

The first dataset we consider is INaturalist which contains slightly more than $500K$ training samples for its mini version (the one commonly employed, see *e.g.* [14]). It contains almost $10K$ actual classes and most SSL methods focus on transfer learning *e.g.* transferring with a Resnet50 from Imagenet-1k lead to SimCLR's 37.2%, MoCoV2's 38.6, BYOL's 47.6 and BarlowTwins'46.5. However training on INaturalist directly produces lower performances reaching only 29.1 with MSN and a ViT. Using DIET is possible out-of-the-box with Resnet18 and ViT variants as their embedding is of dimension

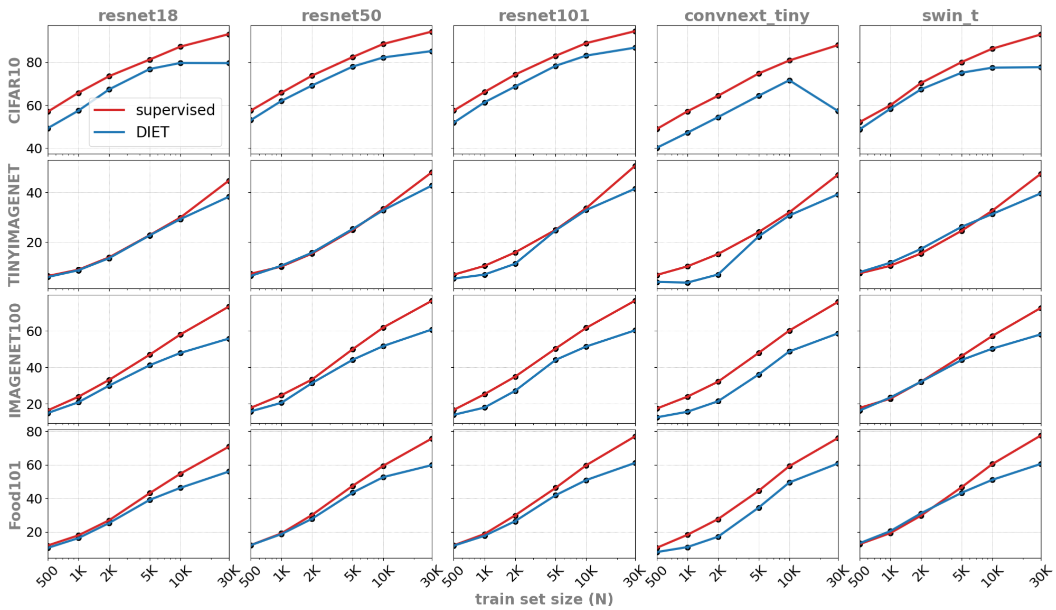


Figure 7: Reprise of Appendix D on additional datasets depicting how DIET is able to compete with supervised learning for in-distribution generalization in very small dataset regime.

512 and 762 respectively making $\boldsymbol{W}$ fit in memory. We obtain 22.81 with a convnext small, and 21.6 with a ViT.

The second dataset we consider is the full Imagenet-1k dataset which contains more than 1 million training samples and 1000 actual classes. In this case, it is not possible to directly hold $\boldsymbol{W}$ in-memory. We however tried a simple strategy which simply consists of sub-sampling the training set to a more reasonable size. This means that although we are putting aside many training images, we enable single GPU Imagenet training with DIET. With a training size of $400K$, we able to reach 44.05 with a convnext small, 43.78 with a SwinTiny, and 44.89 with a ViT/B/16. A standard SSL pipeline has performances ranging between $64\%$ and $72\%$. From those experiments, it is clear that DIET's main limitation comes from very large training set sizes. Although the above simple strategy offers a workable solution, it is clearly not sufficient to match with existing unsupervised learning method and thus should require further consideration. As highlighted in Section 4 below, this is one key avenue for future work.

# G Impact of Training Time and Label Smoothing

In Figure 8 we show the performance of DIET on CIFAR100 across three label smoothing settings. We find higher values of label smoothing speed up convergence, although in this setting all cases greatly benefit from longer training schedules; final linear probe performances are reported in Table 3.

# H Impact of Mini-Batch Size

We show in 9 ablations for TinyImagenet using DIET. In addition we show DIET"s robustness to batch size by conducting an additional ablation by varying the batch size for the Derma MedMNIST dataset with batch sizes as low as 8. As shown in Table 6, we see DIET performs well even with very small batch sizes.

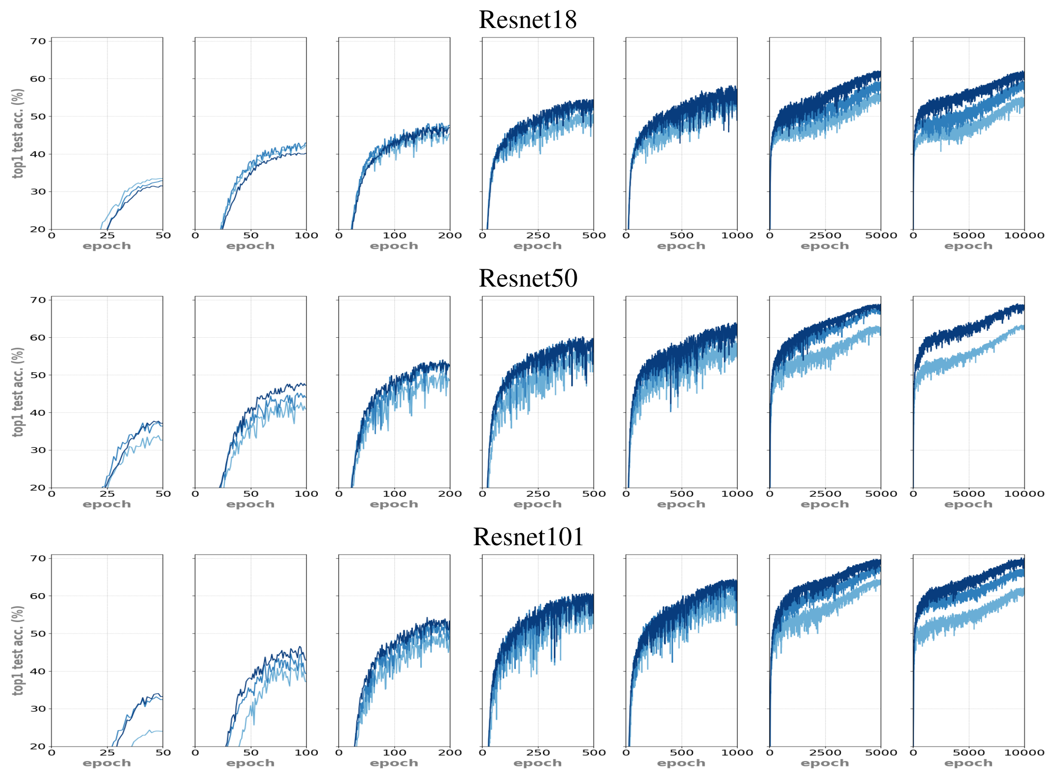


Figure 8: Depiction of the evolution of linear top1 accuracy throughout epochs on CIFAR100 with three Resnet variants and three label smoothing parameters represented by the different **shades of blue** going from light to dark shades with values of $0.1$, $0.4$, and $0.8$ respectively.

| Batch Size | 8 | 32 | 64 | 128 | 512 |
|---|---|---|---|---|---|
| DIET | 71.87 | 72.52 | 73.07 | 74.36 | 71.02 |
| MoCov2 | 66.88 | 64.64 | 66.73 | 66.88 | 61.40 |
| SimCLR | 63.14 | 66.43 | 66.83 | 66.88 | 66.83 |
| VICReg | 65.84 | 60.45 | 64.79 | 66.78 | 66.88 |

Table 6: Reprise of Table 3: DIET's performance across varying batch sizes on the Derma MedMNIST dataset with all other hyperparameter fixed demonstrating the stability of DIET do that hyperparameter and across training iterations. All models are trained for 500 epochs.

# I Impact of Data-Augmentation

To further study the effect of data augmentation in DIET we study varying data augmentation strengths for TinyImageNet in Fig. 9. We also examine the effect of weaker data augmetnations for smaller medical images using PathMNIST in Table 7.

# J DIET compared to supervised learning

**DIET matches supervised learning on datasets with only a few samples per class.** In Appendix D we directly compare DIET with supervised learning on a variety of models and datasets but with controlled training size. We clearly observe that for small dataset, *i.e.*, for which we only use a small part of the original training set (less than 30 images per class), DIET's learned representation is as efficient as the supervised one for the in-distribution classification downstream task.

**DIET works with scattering network architectures** As an additional test, scattering networks [105, 106] hard-code part of the model parameters to be wavelet filter-banks. That specification naturally makes such scattering networks very competitive for small data regimes since the number of

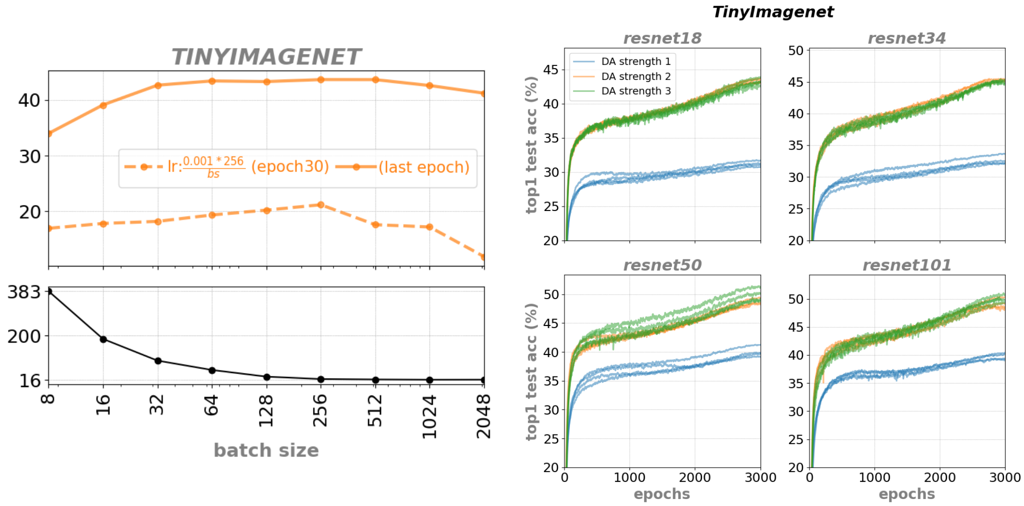


Figure 9: **Left:**TinyImagenet with fixed number of epochs and a single learning rate which is adjusted for each case using the LARS rule therefore per batch-size learning cross-validation can only improve performances, see Table 3, , the per-epoch time includes training, testing, and checkpointing. **Right:** TinyImagenet, see Table 3 for table of results, and the specific DAs can be found in Algorithm 3.

**Algorithm 3** Custom dataset to obtain the indices ($n$) in addition to inputs $\boldsymbol{x}_n$ and (optionally) the labels $y_n$ to obtain `train_loader` used in **??** (Pytorch used for illustration).

```
transforms = [
    RandomResizedCropRGBImageDecoder((size, size)),
    RandomHorizontalFlip(),
]
if strength > 1:
    transforms.append(
        T.RandomApply(
            torch.nn.ModuleList([T.ColorJitter(0.4, 0.4, 0.4, 0.2)]), p=0.3
        )
    )
    transforms.append(T.RandomGrayscale(0.2))
if strength > 2:
    transforms.append(
        T.RandomApply(
            torch.nn.ModuleList([T.GaussianBlur((3, 3), (1.0, 2.0))]), p=0.2
        )
    )
    transforms.append(T.RandomErasing(0.25))
```

degrees of freedom is reduced. We therefore performed two additional experiments: Training a hybrid scattering network in a supervised setting Training a hybrid scattering network with DIET and then learning a linear probe on top (keeping the hybrid scattering frozen) We perform both cases above on the full CIFAR10 training set and on a reduced training set of 5000 (10% of the training data) samples. Supervised training of the scattering network results in 72.1% (58.2%) test set accuracy, whereas unsupervised DIET pretraining followed by a linear probe results in 77.64% (62.8%) for the same architecture. From that experiment we obtain two novel insights. First, DIET works out-of-the-box on DNs such as the hybrid scattering network, with a reduced number of parameters. Second, even in that regime, DIET provides strong performances.

## K Additional Results for MedMNIST

In Figure 10 we show training curves for DIET with a ResNet18 architecture. We perform additional experiments with DIET using a vision transformer architecture (ViT-Small with patch size 4) based on the architecture from `https://github.com/lucidrains/vit-pytorch/blob/main/vit_pytorch/vit_for_small_dataset.py`. We find DIET achieves good performance on the same MedMNIST datasets with this ViT architecture without additional hyperparameter tuning as shown in Table 8 and in comparison to all three baseline SSL methods in Table 7.

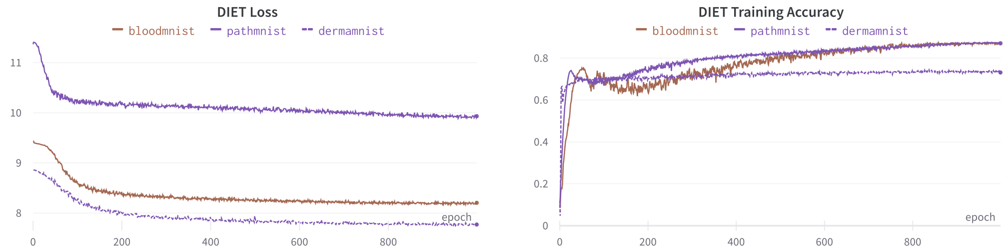


Figure 10: DIET MedMNIST training loss curves for the DIET criterion (left) and training accuracy (right) with a ResNet18 backbone.

| | **bloodmnist** | | **dermamnist** | | **pathmnist** | |
|---|---|---|---|---|---|---|
| | **train** | **test** | **train** | **test** | **train** | **test** |
| **DIET** | 77.65 | 81.85 | 71.03 | 68.88 | 56.37 | 21.27 |
| **SimCLR** | 82.48 | 79.45 | 69.13 | 32.37 | 69.45 | 21.80 |
| **VICReg** | 86.71 | 81.03 | 69.89 | 46.33 | 82.94 | 12.76 |
| **MoCov2** | 62.76 | 51.01 | 66.78 | 63.39 | 72.9 | 41.75 |

| DIET | PathMNIST | |
|---|---|---|
| Augmentation | train | test |
| Default | 56.37 | 21.27 |
| Weak | 44.90 | 48.95 |
| None | 44.65 | 45.67 |

Table 7: **Top:**DIET performance across the three MedMNIST datasets using a transformer (ViT-S) architecture with patch size 4 in comparison to standard SSL baselines with the same ViT architecture. **Bottom:**Comparing DIET's performance across data augmentations for PathMNIST using a transformer (ViT-S) architecture with patch size 4. Weak augmentation corresponds to only random resized cropping and horizontal flipping.

Table 8: DIET performance across the three MedMNIST datasets using a transformer (ViT-S) architecture with patch size 4. In the first row we show the performance of a baseline SimCLR model with the default ResNet18 encoder for comparison.

| dataset | bloodmnist | | dermamnist | | pathmnist | |
|---|---|---|---|---|---|---|
| | train | test | train | test | train | test |
| DIET | 77.65 | 81.85 | 71.03 | 68.88 | 56.37 | 21.27 |

We find evidence of the default augmentations for PathMNIST being too aggressive and confirm DIET's performance improves with the use of weaker augmentations in Table 7. Surprisingly, we find DIET performs quite well with no augmentations at all, a setting in which most standard SSL methods would be impossible to train.

# 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: [NA]

   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: [NA]

   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: [Yes]

Justification: [NA]

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: [NA]

   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: [NA]

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: [NA]

   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: [No]

   Justification: We do not report eror bars, but instead carefully study and report the stability of our results across various hyperparameter and architecture choices to make clear the results are not an artifact of stochasticity during training. For baselines, we report numbers from publicly available papers when possible, which we found often lack error bars as well.

   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: [NA]

   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: [NA]

   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: [Yes]

    Justification: [NA]

    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: [NA]

    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: [NA] .

    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: [NA]

    Justification: [NA]

    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: [NA]

    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: [NA]

    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.