ON THE SURPRISING EFFICACY OF DISTILLATION AS AN ALTERNATIVE TO PRE-TRAINING SMALL MODELS

Sean Farhat & Deming Chen
Department of Computer Science
University of Illinois, Urbana-Champaign
{seanf2, dchen}@illinois.edu

ABSTRACT

In this paper, we propose that small models may not need to absorb the cost of pre-training to reap its benefits. Instead, they can capitalize on the astonishing results achieved by modern, enormous models to a surprising degree. We observe that, when distilled on a task from a pre-trained teacher model, a small model can achieve or surpass the performance it would achieve if it was pre-trained then finetuned on that task. To allow this phenomenon to be easily leveraged, we establish a connection reducing knowledge distillation to modern contrastive learning, opening two doors: (1) vastly different model architecture pairings can work for the distillation, and (2) most contrastive learning algorithms rooted in the theory of Noise Contrastive Estimation can be easily applied and used. To illustrate these points, we demonstrate this paradigm using pre-trained teacher models from open-source model hubs, Transformer and convolution based model combinations, and a novel distillation algorithm that massages the Alignment/Uniformity perspective of contrastive learning by Wang & Isola (2020) into a distillation objective. We choose this flavor of contrastive learning due to its low computational cost, an overarching theme of this work. We also observe that this phenomenon tends not to occur if the task is data-limited. However, this can be alleviated by leveraging yet another scale-inspired development: large, pre-trained generative models for dataset augmentation. Again, we use an open-source model, and our rudimentary prompts are sufficient to boost the small model's performance. Thus, we highlight a training method for small models that is up to 94% faster than the standard pre-training paradigm without sacrificing performance. For practitioners discouraged from fully utilizing modern foundation datasets for their small models due to the prohibitive scale, we believe our work keeps that door open.

1 INTRODUCTION

Small machine learning models are incredibly valuable: cost savings on memory and compute, reductions on energy footprints, and deployment on resource-constrained edge devices. However, recent attention has focused on training immense models on immense datasets, dubbed "foundation models" (Radford et al., 2021; 2019; Brown et al., 2020; Chen et al., 2020a; Bommasani et al., 2021; Alayrac et al., 2022; Yuan et al., 2021). Their surprisingly high performance and generalization capabilities has led to the current motivating sentiment of unimpeded scaling. How can small models keep up?

Various lines of research tackle how to squeeze out maximal performance from small models, all targeting different points in the training process. Before training, neural architecture search (NAS) (Zoph & Le, 2017; Liu et al., 2019) algorithmically constructs small yet high-performing model designs. After training, one can prune or quantize (Jacob et al., 2017; Han et al., 2016) a model to reduce its size with a minimal performance drop. We focus on what happens in the middle: the training process. During training, knowledge distillation (KD) Bucilua et al. (2006); Hinton et al. (2015) can assist the model by having it also learn from a larger teacher model that is an expert at the task.

We consider the situation of taking the main strategy that defines foundation models' success, pre-training, and applying it to small models. This has been a classic strategy for boosting the performance of models in general, however the modern scale of foundation datasets growing to billions of samples Schuhmann et al. (2022) puts pre-training at odds with the main appeal of small models: low cost at both train and inference time. One can hope that a pre-trained small model exists in a public model hub, however what if a custom model architecture is desired? Then, a choice must be made: either (a) absorb the cost of pre-training and reap the performance benefits, or (b) train from scratch on just the desired task, absorbing the probable reduction in performance that comes with decreasing model size. Addressing this dilemma, we re-evaluate if pre-training is necessary for small models. Often, the demand for small models stems from deploying them for one or a handful of tasks. Therefore, does a small model need a *comprehensive* feature backbone? Alternately, what if we teach it to behave like it was pre-trained and finetuned, but only on the relevant slice of knowledge (see Figure 1)?

In this paper, we show that, by leveraging the progress in foundation models, contrastive learning, and pre-trained generative models, small models can achieve and *surpass* pretrain-then-finetune performance without ever needing to touch a pre-training dataset. Our approach is simple yet effective: (1) using *any* publicly available teacher pre-trained on an appropriate foundation dataset, optionally finetune it for the desired task, then (2) transfer its knowledge of the task via our proposed distillation loss on (3) the task dataset augmented with synthetic samples generated from publicly available pre-trained generative models. Our method can easily be applied to improve performance for any small model on any task, can be used in continual learning regimes (Li & Hoiem, 2017) if multiple tasks are desired, and fits in nicely between methods like NAS and pruning to fully utilize all the tools to maximize small model accuracy.

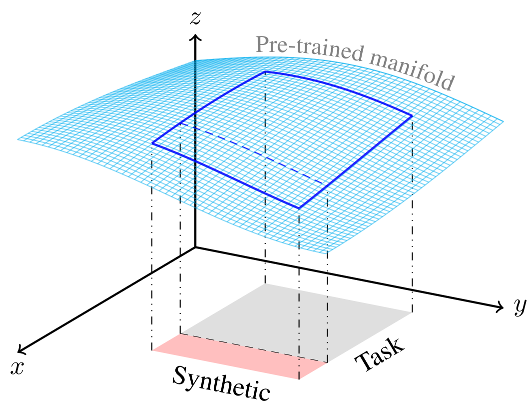


Figure 1: By only mimicking the relevant slice of a pre-trained manifold, a small model can achieve the *same or better* performance than if it had been fully pre-trained and finetuned. Adding synthetic samples leads to better generalization.

We test our approach on 6 visual recognition tasks, spanning both the data-abundant and data-limited regimes, the latter of which benefits the most from pre-training. The formulation of our KD loss as a contrastive objective allows our setup to be agnostic to the underlying teacher-student architectures, a luxury rarely granted from prior work on distillation algorithms. We demonstrate this by utilizing teachers representative of the main vision architectures, a Vision Transformer (Dosovitskiy et al., 2020) and 50-layer ResNet (He et al., 2016), to assist 2 small students, a MobileNetV2 (Sandler et al., 2018) and 18-layer ResNet. We also provide a cost analysis of our method: skipping pre-training leads to large resource benefits by reducing overall training time. However, a nontrivial image generation cost is added given the state of the diffusion model we employ. Our code can be found at `https://github.com/sfarhat/dapt`.

2 Related Work

Knowledge Distillation Knowledge Distillation (KD) first appeared in the realm of ensembling methods (Dietterich, 2000), though it was Bucilua et al. (2006) who first used $n \geq 1$ expert models to teach a smaller model, thus considering their technique a form of model compression. Hinton et al. (2015) popularized the method by using a temperature-based softmax. Since then, KD has blossomed into an active area of research. Logit-Based KD algorithms (Hinton et al., 2015; Zhao et al., 2022; Yang et al., 2021; Chen et al., 2022) look at the logits, the input to the classification softmax. Feature-based algorithms (Romero et al., 2014; Zagoruyko & Komodakis, 2016; Ahn et al., 2019; Passalis & Tefas, 2018; Heo et al., 2019b; Kim et al., 2018; Huang & Wang, 2017; Heo et al., 2019a; Chen et al., 2021a;b; Miles et al., 2021; Srinivas & Fleuret, 2018; Tian et al., 2020) probe the middle layers of both networks and take different perspectives on how intermediate knowledge

Contrastive-Based Backbone Matching   Augmented Task Dataset

$$\min_{\theta_S} \mathcal{L}_{\text{CE}}(f_S, \mathcal{D}) + \underbrace{\mathcal{L}_{\text{Embed}}(f_S, f_T, \mathcal{D}) + \mathcal{L}_{\text{KD}}(f_S, f_T, \mathcal{D})}_{\text{Mimic pre-trained teacher}}$$

Supervised Learning

Method	High Accuracy	Low Cost
Supervised	✗	✓
Pre-training	✓	✗
Ours	✓	✓

Figure 2: (Left) Our proposed alternative to pre-training and finetuning. (Right) The benefits of our method compared to standard approaches.

should be quantified and transferred. Relation-based methods (Park et al., 2019; Liu et al., 2021; Yim et al., 2017; Tung & Mori, 2019; Peng et al., 2019) look at the inter-batch relationships between outputs of the teacher and student networks. Tian et al. (2020) were the first to apply the idea of contrastive learning to KD, via a formulation inspired by Wu et al. (2018), though their work pre-dated most modern advancements in contrastive learning. The target performance we aim to beat, a pre-trained student, has rarely been addressed. Most prior work looked at distilling all of ImageNet (Deng et al., 2009; Russakovsky et al., 2015) and finetuning the student after to test task performance.

Contrastive Learning Contrastive Learning comes from the idea of Noise Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010). Broadly speaking, the main idea is to pull "positive pairs" together, and push "negative pairs" far apart. Early iterations of this idea (Chopra et al., 2005; Schroff et al., 2015; Sohn, 2016; Salakhutdinov & Hinton, 2007; Frosst et al., 2019; Oord et al., 2018) dealt with different sources and quantities for positive and negative pairs. Most modern advancements in contrastive learning come from the realm of self-supervised learning (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; Grill et al., 2020; Chen & He, 2021; Zbontar et al., 2021).

Generative Models as Data Due to the increased realism of synthetic images, recent work has examined if they are good enough to use as a source of data since we can infinitely sample from generative models. Currently, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020), equivalent to score-based methods (Hyvärinen, 2005; Vincent, 2011; Song & Ermon, 2019; Song et al., 2020) offer the best generative guarantees in terms of sampling diversity and likelihood maximization. The only downside to diffusion models, as of writing, is their slow sampling speed. However, improving sampling speeds with minimal effect on sample quality is an active area of research (Song et al., 2023; Salimans & Ho, 2022). The works of He et al. (2022); Zhou et al. (2023); Bansal & Grover (2023); Sariyildiz et al. (2022); Azizi et al. (2023) used diffusion to augment the training dataset for supervised training with positive results, but did not consider the KD setting.

3 Method

Our main idea is to assist a small model's supervised learning process for task $\mathcal{T}$ by having it match the behavior of a large "teacher" model f_T, which has been pre-trained then finetuned for $\mathcal{T}$. In doing so, the target small "student" model f_S behaves as if it too was pre-trained and finetuned on $\mathcal{T}$, when in reality, it never touches the pre-training dataset. Our approach builds on the standard KD paradigm by (1) deliberately using a pre-trained f_T, (2) introducing a simple, inexpensive, and architecture-agnostic distillation algorithm deeply linked to contrastive learning, and (3) using supplementary synthetic data as needed.

3.1 Using a Pre-Trained Teacher f_T

In order for f_S to behave like it was pre-trained, it should learn from a teacher f_T that was. Model hubs make these teachers easily accessible. While not necessary, we can additionally finetune f_T on τ to add an extra learning signal for the student. To reflect modern practices, when we refer to finetuning, we imply the method of *linear probing* (LP) which freezes the backbone weights and only updates the task-specific head.

3.2 THE CONTRASTIVE-BASED EMBEDDING LOSS

We want to ensure the student's backbone mimics a pre-trained one on the desired task. To achieve this and avoid the complexities induced by the implicit biases of model architectures, a common pitfall in prior work, we look to contrastive learning theory, which only utilizes the final embeddings h.

Contrastive-Based Distillation For the sake of simplicity, in this section we will ignore the classification heads and view the student f_S and teacher f_T as models that generate high-dimensional representations $h_S \in \mathbb{R}^S, h_T \in \mathbb{R}^T, S \neq T$ after their final backbone layer (Figure 4b). We believe a robust measure of distance can be effective given its recent success in the self-supervised domain: contrastive learning. Following the ideas from their literature, we append "projection" modules $g_S : \mathbb{R}^S \to \mathbb{R}^d, g_T : \mathbb{R}^T \to \mathbb{R}^d$, where $d \ll S, T$. Thus, given a data distribution $x \sim p_{\text{data}}$, our models induce distributions $p_S(\cdot), p_T(\cdot)$, such that $g_S \circ f_S(x) =: z_S \sim p_S, g_T \circ f_T(x) =: z_T \sim p_T$. Note that both $g(\cdot)$'s include a final normalization step that ensures $||z||_2 = 1$. We also define the distribution of "positive" pairs $p_{\text{pos}}(\cdot, \cdot)$, where the marginals should match: $\forall z_S, \int p_{\text{pos}}(z_S, z_T) dz_T = p_S(z_S)$ and $\forall z_T, \int p_{\text{pos}}(z_S, z_T) dz_S = p_T(z_T)$. We approximate sampling from $p_{\text{pos}}(\cdot, \cdot)$ by passing the same sample $x_i \sim p_{\text{data}}$ through both networks and projectors to generate the positive pair $(z_{S,i}, z_{T,i})$. *Any* sample $x_j, j \neq i$ that creates a $z_{\cdot,j}$ induces negative pairs $(z_{S,i}, z_{T,j}), (z_{S,j}, z_{T,i})$.

At its core, contrastive learning pulls positive pairs together and pushes negative pairs apart. Our proposed formulation that achieves this, inspired by the InfoNCE loss of Oord et al. (2018), is to minimize the following with respect to the parameters of the student f_S and both projectors g_S, g_T:

$$\mathbb{E}_{(z_{S,i}, z_{T,i}) \sim p_{\text{pos}}, \{z_{S,j}\}_{j=1}^M \sim p_S, \{z_{T,j}\}_{j=1}^M \sim p_T} \left[-\log \frac{e^{{z_{S,i}}^\top z_{T,i}/\tau}}{\sum_j e^{{z_{S,i}}^\top z_{T,j}/\tau} + \sum_j e^{{z_{S,j}}^\top z_{T,i}/\tau}} \right] \quad (1)$$

where τ is a temperature hyperparameter and M is the number of negative samples chosen beforehand. This formulation allows us to reduce any contrastive-based objective to a distillation one. We choose one that is inexpensive and interpretable as an illustration, leaving other contrastive-distillation collaborations to future work.

Optimizing Alignment and Uniformity Wang & Isola (2020) showed that as the number of negative samples $M \to \infty$, then 2 simpler quantities can be optimized instead: the *alignment* (cosine similarity) of the positive pairs, and the *uniformity* of all (normalized) samples on the hypershpere in $\mathbb{R}^d$. With no negative sample bank, momentum model, or large batch size, this method is the most lightweight contrastive algorithm. Thus, we chose Alignment/Uniformity (**A/U**) as the core of our contrastive-distillation algorithm. We re-express their metrics to reflect our distillation flavor of generating positive pairs:

$$\mathcal{L}_{\text{align}} := \mathbb{E}_{(z_{S,i}, z_{T,i}) \sim p_{\text{pos}}}[||z_{S,i} - z_{T,i}||_2^\alpha] \qquad \mathcal{L}_{\text{uniform}} := E_{(z_{Z,i}, z_{Z,j}) \sim p_Z}[e^{-t||z_{Z,i} - z_{Z,j}||_2^2}] \quad (2)$$

where $\alpha > 0, t > 0, Z \in \{S, T\}$. We can combine these into one term: $\mathcal{L}_{\text{Embed}} := w_{\text{align}} \cdot \mathcal{L}_{\text{align}} + w_{\text{uniform}} \cdot \mathcal{L}_{\text{uniform}}$. and use the default parameters suggested in Wang & Isola (2020): $w_{\text{align}} = 1, w_{\text{uniform}} = 1, \alpha = 2, t = 2$. Pseudocode can be found in Figure 4a.

3.2.1 THE FINAL TRAINING OBJECTIVE

If we finetune f_T, we can leverage the original method of Hinton et al. (2015) and add their effective logit-based loss $\mathcal{L}_{KD}$ (Equation 3). Altogether, we can combine the above loss functions with the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ on the true one-hot labels: $\mathcal{L} := \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{Embed}} + \lambda_3 \mathcal{L}_{\text{KD}}$. We used $\lambda_1 = \lambda_2 = \lambda_3 = 1$. Ablations for the overall A/U design can be found in Appendix A.4.

3.3 AUGMENTING THE TRANSFER DATASET

Lastly, we address the transfer dataset $\mathcal{D}$. Due to the high costs, we avoid touching the pre-training dataset, but *only* using the task dataset is suboptimal. We cannot deny the power of more data, so we propose to apply the known strategy of data augmentation to the distillation setting. Given the recent successes of diffusion models in generative modelling, we choose to use a pre-trained text-to-image model, Stable Diffusion (Rombach et al., 2022), as our source of extra samples. We leverage the publicly available stable-diffusion-v1-4[1], which was pre-trained on LAION-2B (Schuhmann et al., 2022), and guide its generation with the appropriate language prompts (Appendix A.3).

4 EXPERIMENTS

Setup We choose data-limited datasets since they are most helped by pre-training: MIT-67 (Quattoni & Torralba, 2009), CUB-2011 (Wah et al., 2011), DTD (Cimpoi et al., 2014), and Caltech-101 (Li et al., 2003). In addition, we evaluate on standard vision benchmarks: CIFAR-10 and CIFAR-100 Krizhevsky et al. (2009). We test our method with 2 teachers: a ResNet50 (He et al., 2016) and a base Vision Transformer (Dosovitskiy et al., 2020), ViT-B-16. For the target small models, we choose a ResNet18 and a MobileNetV2 (Sandler et al., 2018). Details about optimizers, schedulers, and input augmentations can be found in Appendix A.5.

Results We keep our main observations in this section and provide more detailed experiments in Appendices A.1.1, A.1.2, and A.1.3. The baselines we compare to are when the small model was either (1) pre-trained on ImageNet for 100 epochs and linearly probed for the task or (2) trained on the task from scratch. We denote augmenting the transfer dataset with synthetic samples as "A/U ($n\times$)", where $n\times$ indicates the number of extra samples in terms of the size of the respective train set (see Appendix A.6). We only use synthetic data in the case where the transfer dataset is so limited that A/U ($0\times$) lags too far behind the pre-trained goal. In addition, we apply standard image augmentations, e.g. manipulating color properties, cropping, and flipping. The results of the ResNet50-MobileNetV2 pair can be found in Figure 3. Other pairings can be found in Appendix A.1.3. The cost of A/U ($n\times$) includes the times to finetune the teacher and generate the extra images. As we can see, while maintaining a competitive or superior accuracy to their pre-trained counterpart, our method can cut training time by up to 94%, though gets slower if more samples need to be generated. We do not include the time to pre-train the teacher or generative models since those are assumed to be obtained already trained off-the-shelf.

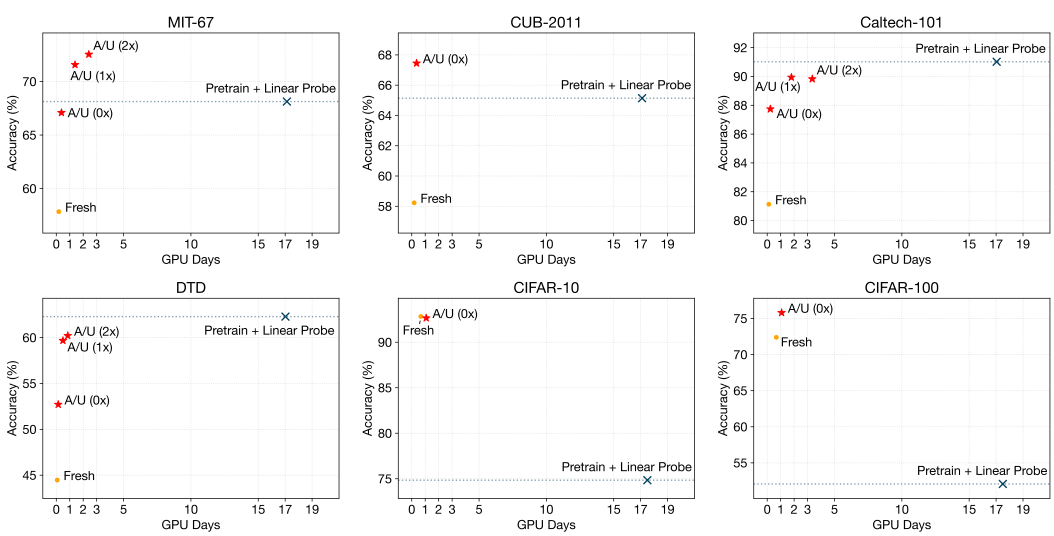


Figure 3: Cost/accuracy comparison of our method (A/U) to supervised training (Fresh) and pre-training then linear probing. The teacher was a ResNet50 and the student was a MobileNetV2. All timing experiments were done on one NVIDIA P100 GPU.

[1]`https://huggingface.co/CompVis/stable-diffusion-v1-4`

5 Conclusion

Mainstream attention continues to focus on large models. Fortunately, we show that small models can leverage these advancements for their own benefit. By combining a pre-trained teacher, a novel knowledge transfer algorithm, and augmenting the "questions" the student asks the teacher, any small model can significantly improve its performance without weeks of pre-training; all that is required is a publicly available, pre-trained teacher, a contrastive learning algorithm, and a generative model. The demand for small models will always exist due to resource constraints or savings, and when combined with methods like NAS and pruning, small models have a bright future.

Acknowledgments

This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign (Kindratenko et al., 2020). SF was supported by the IBM-Illinois Discovery Accelerator Institute and the AMD Center of Excellence at the University of Illinois at Urbana-Champaign.

References

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9163–9171, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 535–541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL `https://doi.org/10.1145/1150402.1150464`.

Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7028–7036, 2021a.

Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11933–11942, 2022.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021b.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

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

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Thomas G. Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, 2000.

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.

Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. {SEED}: Self-supervised distillation for visual representation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=AHm3dbp7D1D`.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pp. 2012–2020. PMLR, 2019.

Yuting Gao, Jia-Xin Zhuang, Shaohui Lin, Hao Cheng, Xing Sun, Ke Li, and Chunhua Shen. Disco: Remedy self-supervised learning on lightweight models with distilled contrastive learning. *arXiv preprint arXiv:2104.09124*, 2021.

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.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/gutmann10a.html`.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016.

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*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019a.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3779–3787, 2019b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL `http://jmlr.org/papers/v6/hyvarinen05a.html`.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference, 2017.

Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018.

Volodymyr Kindratenko, Dawei Mu, Yan Zhan, John Maloney, Sayed Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, and William Gropp. Hal: Computer system for scalable deep learning. In *Practice and experience in advanced research computing*, pp. 41–48. 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009.

FF Li, M Andreetto, MA Ranzato, and P Perona. Caltech 101. *Computational Vision Group, California Institute of Technology*, 7, 2003.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search, 2019.

Weidong Liu, Shibo Nie, Junhui Yin, Rui Wang, Donghui Gao, and Ling Jin. Sskd: Self-supervised knowledge distillation for cross domain adaptive person re-identification. In *2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*, pp. 81–85. IEEE, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Roy Miles, Adrian Lopez Rodriguez, and Krystian Mikolajczyk. Information theoretic representation distillation. *arXiv preprint arXiv:2112.00459*, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5007–5016, 2019.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 413–420. IEEE, 2009.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pp. 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL https://proceedings.mlr.press/v2/salakhutdinov07a.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning (s) from a synthetic imagenet clone. *arXiv preprint arXiv:2212.08420*, 2022.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning*, pp. 4723–4731. PMLR, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 07 2011. ISSN 0899-7667. doi: 10.1162/NECO_a_00142. URL https://doi.org/10.1162/NECO_a_00142.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

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

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018.

Haohang Xu, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, Xinggang Wang, Wenrui Dai, Hongkai Xiong, and Qi Tian. Bag of instances aggregation boosts self-supervised distillation. *arXiv preprint arXiv:2107.01691*, 2021.

Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *ICLR2021*, 2021.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7130–7138, 2017. doi: 10.1109/CVPR.2017.754.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. 2023.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning, 2017.

A Appendix

A.1 Additional Results and Model Pairings

In our results, we use several abbreviations for brevity. **FR** stands for a model that was initialized randomly and trained end-to-end on the task. **LP** refers to a model that was pre-trained on ImageNet (Deng et al., 2009; Russakovsky et al., 2015) and linearly probed for the task. **T** and **S** refer to the teacher and student models, respectively. Note that all non-A/U trainings (i.e. finetuning the teacher, vanilla training the student), were done with just the dataset; no synthetic samples were involved.

A.1.1 Self-Supervised Transfer Learning

In the self-supervised learning literature, pre-training is significantly less effective as model size decreases. SEED (Fang et al., 2021), DisCo (Gao et al., 2021), and BINGO (Xu et al., 2021) attempt to mitigate this by augmenting training with distillation from a pre-trained teacher backbone. They test performance by training/distilling on ImageNet first, then adapting to tasks. For us, we sidestep ImageNet and run our distillation directly on the downstream task. We compare our strategy to theirs in Table 1. We use the same teacher model, a pre-trained MoCoV2 ResNet50, and student, a ResNet18. The baseline is a ResNet18 pre-trained on ImageNet via MoCoV2 and linearly probed for a task. We omit the teacher's head to match their teacher-student setup ($\lambda_3 = 0$), so the knowledge transfer is purely feature-based. Our method surpasses prior work by a large margin, lending support to the idea that small models may not need their own strong backbone, especially in the self-supervised domain where model size matters more.

Table 1: Comparing to self-supervised distillation.

Method	CIFAR-10	CIFAR-100
Baseline	77.9	48.1
SEED (Fang et al., 2021)	82.3	56.8
DisCo (Gao et al., 2021)	85.3	63.3
BINGO (Xu et al., 2021)	86.8	66.5
A/U (Ours)	**94.33**	**73.83**

A.1.2 Accuracy Comparison

We provide a tabular visualization of how our method compares to pre-training in Table 2. On 4 out of 6 datasets, we are able to surpass the pre-training performance, while on the remaining two, which are considered quite data-limited, we come within 2% when using the added synthetic data.

Table 2: Our method (A/U) can achieve performance close to, and often surpass, pre-training.

Method	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
A/U (0×)	**92.66**	**75.8**	67.09	**67.45**	87.74	52.71
A/U (1×)	-	-	71.57	-	89.94	59.68
A/U (2×)	-	-	**72.54**	-	89.83	60.21
S-LP	74.84	52.08	68.13	65.14	**91.02**	**62.29**

A.1.3 Comparison to Other Methods on Various Model Pairings

For a fair comparison, we also test using a pre-trained teacher with prior state-of-the-art KD algorithms that did not consider our setting. Notably, we choose ones that have demonstrated high efficacy as well as being architecture-agnostic; they do not utilize any intermediate feature information. The three works we compare to are:

1. KD (Hinton et al., 2015): The original KD loss that compares logits.

2. CRD++ (Tian et al., 2020): A contrastive-based distillation method. We also include $\mathcal{L}_{\text{KD}}$. As their work pre-dated many modern advancements in contrastive learning, we compare our method to a modernized version which substitutes the linear projection $g(\cdot)$ with the same 2-layer MLP we use in our A/U loss.

3. SRRL (Yang et al., 2021): A method that passes the student features through the teacher classifier. We do not use $\mathcal{L}_{\text{KD}}$ here as the original authors did not. However, they only considered convolutional networks and utilized a small convnet to project the student features to the teacher's dimension. We substitute this with an MLP to make it general to the underlying model architecture (Figure 6).

The implementations of these methods were taken from their open-source repositories, which we thank the authors for making available, and adapted appropriately. These results can be found in Tables 3, 4, 5, and 6. In all setups, our algorithm (A/U) leads to high performance gains that are competitive with, and sometimes surpass, prior work. On one hand, these results demonstrate the superiority of our A/U loss in certain setups; on the other hand, many existing KD algorithms can be substituted in $\mathcal{L}$ for our overall paradigm and still offer competitive boosts to small model performance. We also emphasize that, while CRD++ is also a contrastive method, our A/U method is computationally cheaper due to the lack of a negative sample bank, while often outperforming it.

Table 3: Accuracies (%) of MobileNetV2 when assisted by a pre-trained ResNet50.

Method	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
T-LP	77.3	53.65	72.39	63.26	92.94	65.96
S-FR	92.83	72.39	57.84	58.23	81.13	44.47
KD	92.92	75.97	66.04	**67.48**	86.61	52.02
SRRL	92.85	73.24	65.67	62.81	85.31	49.79
CRD++	**93.45**	**76.37**	65.75	67.21	86.89	52.45
A/U (Ours)	92.66	75.8	**67.09**	67.45	**87.74**	**52.71**
S-LP	74.84	52.08	68.13	65.14	91.02	62.29

Table 4: Accuracies (%) of MobileNetV2 when assisted by a pre-trained ViT-B-16.

Method	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
T-LP	95.14	80.56	81.34	79.06	94.80	70.21
S-FR	92.83	72.39	57.84	58.23	81.13	44.47
KD	94.33	78.9	**66.34**	**71.16**	84.92	**49.79**
SRRL	93.99	78.66	63.51	67.66	**85.31**	49.73
CRD++	93.93	78.57	65.37	70.45	82.49	48.4
A/U (Ours)	**94.55**	**79.71**	65.97	70.87	84.8	49.47
S-LP	74.84	52.08	68.13	65.14	91.02	62.29

Table 5: Accuracies (%) of ResNet18 when assisted by a pre-trained ResNet50.

Method	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
T-LP	77.3	53.65	72.39	63.26	92.94	65.96
S-FR	93.22	71.44	49.85	49.17	77.40	34.47
KD	92.48	**75.94**	65.22	**64.77**	85.14	46.44
SRRL	**92.93**	72.06	64.63	62.74	**86.27**	47.82
CRD++	92.87	75.72	65.75	64.19	85.25	**48.78**
A/U (Ours)	92.79	75.11	**66.72**	63.88	85.54	46.01
S-LP	78.38	55.44	66.27	62.75	90.23	62.07

Table 6: Accuracies (%) of ResNet18 when assisted by a pre-trained ViT-B-16.

Method	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
T-LP	95.14	80.56	81.34	79.06	94.80	70.21
S-FR	93.22	71.44	49.85	49.17	77.40	34.47
KD	94.54	78.95	**65.82**	**68.81**	83.28	**44.15**
SRRL	94.03	77.46	61.87	62.94	81.58	37.66
CRD++	93.94	**79.06**	63.28	67.35	79.49	42.61
A/U (Ours)	**94.98**	78.89	65.75	68.48	**83.39**	40.69
S-LP	78.38	55.44	66.27	62.75	90.23	62.07

A.2 DETAILS OF OUR LOSS

Our contrastive-based loss (Equation 1) can be interpreted as translating the InfoNCE loss of Oord et al. (2018) to the distillation case:

$$\mathbb{E}_{(x,y)\sim p_{\text{pos}},\{x_i^-\}_{i=0}^M\sim p_{\text{data}}}\left[-\log\frac{e^{f(x)^\top f(y)/\tau}}{e^{f(x)^\top f(y)/\tau}+\sum_i e^{f(x_i^-)^\top f(y)/\tau}}\right] \quad \text{(InfoNCE)}$$

We also employ the original method of Hinton et al. (2015), which compares the student and teacher logits (pre-softmax vectors) via the cross-entropy:

$$\mathcal{L}_{KD}(z_s, z_t) = \tau^2 \mathcal{H}(\sigma(z_t/\tau), \sigma(z_s/\tau)) \tag{3}$$

where σ is the softmax function and τ is a temperature hyperparameter that controls the smoothness of the softmax; a higher τ leads to a more uniform distribution.

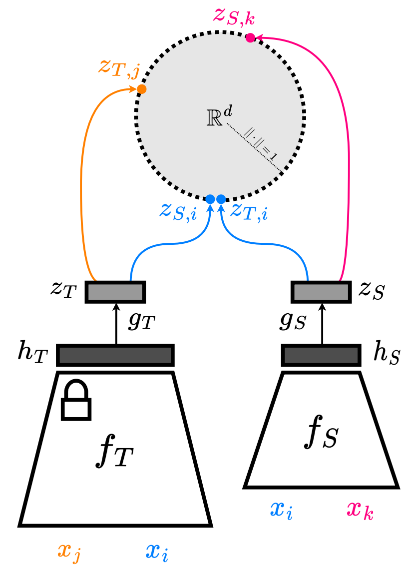
```
# x: batch of samples
# y: batch of other positive half

import torch
def align_loss(x, y, alpha=2):
    dists = (x - y).norm(p=2, dim=1)
    return dists.pow(alpha).mean()

def unif_loss(x, t=2):
    pair = -t*torch.pdist(x, p=2)**2
    return pair.exp().mean().log()
```

Require: f_S, f_T, g_S, g_T: models and projectors
Require: w_a, w_u: loss weights
1: Sample batch x
2: $z_S \leftarrow g_S \circ f_S(x), z_T \leftarrow g_T \circ f_T(x)$
3: $l_a = \texttt{align\_loss}(z_S, z_T)$
4: $l_u = \frac{1}{2}(\texttt{unif\_loss}(z_S) + \texttt{unif\_loss}(z_T))$
5: $\mathcal{L}_{\text{Embed}} = w_a l_a + w_u l_u$

(a) Pseudocode



(b) Flow chart

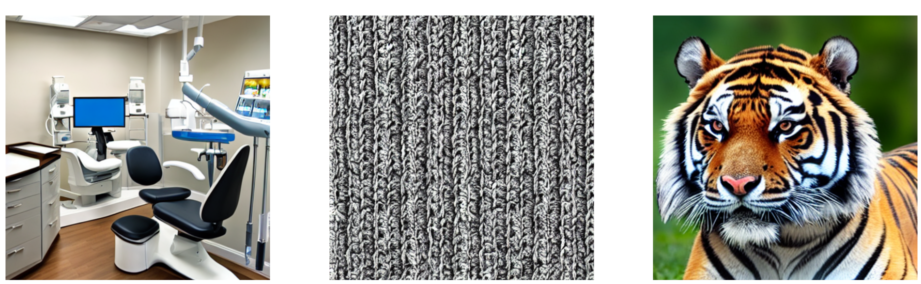
Figure 4: Our Alignment/Uniformity (A/U) based contrastive loss.

A.3 Synthetic Data Generation Details

We find that 3 datasets require synthetic data to get close to pre-train levels of performance: MIT-67 (Quattoni & Torralba, 2009), Caltech-101 (Li et al., 2003), and DTD (Cimpoi et al., 2014), all of which are considered data-limited. The diffusion model was set to take 50 inference steps per batch of images, amounting to around 12.3 seconds per image on 1 NVIDIA P100 GPU. Our choice of prompts and timing numbers can be found in Table 7. Samples of generated images can be found in 5.

Table 7: Details of the synthetic image generation process. {class} corresponds to a class name in the dataset.

Dataset	Prompt	1× Generation Cost (hrs)	2× Generation Cost (hrs)
MIT-67	"the inside of a {class}"	18.355 (80 imgs/class)	36.711 (160 imgs/class)
DTD	"{class} texture"	6.438 (40 imgs/class)	12.876 (80 imgs/class)
Caltech-101	"a picture of a {class}"	29.399 (68 imgs/class)	58.799 (136 imgs/class)



(a) "the inside of a dental office"

(b) "knitted texture"

(c) "a picture of a tiger"

Figure 5: Examples of synthetic samples generated for each dataset, (a) MIT-67, (b) DTD, (c) Caltech-101, with the respective prompt.

A.4 Ablations on Contrastive Loss Design

Our ablation experiments were done in the process of designing the A/U loss formulation. We tested whether the original KD loss (Hinton et al., 2015) or the newer SRRL loss (Yang et al., 2021) should be used for the logit-based loss. In addition, we experimented with the design of the projection module $g(\cdot)$. The teacher-student pairing was ResNet50-MobileNetV2. The results can be found in Table 8.

Table 8: Ablation on different projector architectures and logit-based losses for our A/U loss.

Projector	Logit Loss	CIFAR-10	CIFAR-100	MIT-67	CUB-2011	Caltech-101	DTD
MoCoV2	SRRL	**93.25**	73.29	65.82	63.03	87.23	49.63
MoCoV2	KD	92.66	**75.8**	67.09	67.45	**87.74**	**52.71**
Linear	SRRL	93.18	73.16	66.19	62.08	86.95	50.74
Linear	KD	92.81	**75.8**	**67.54**	**68.16**	87.51	52.07

$\mathcal{L}_{\mathbf{KD}}$ vs. SRRL Loss In SRRL (Yang et al., 2021), instead of comparing the student and teacher logits, the student features are first passed through a connector module (Figure 6) that lifts them to the teacher's feature space then passed through the teacher's classifier. Then, these logits are compared. When comparing using this or $\mathcal{L}_{\text{KD}}$ for the logit-based loss, we found that $\mathcal{L}_{\text{KD}}$ worked better.

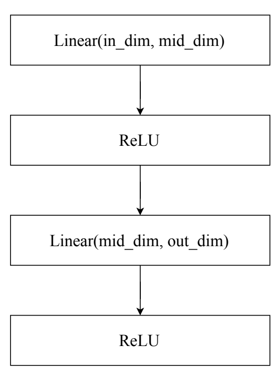


Figure 6: Our modified design of the module that projects the student features to the teacher dimensions in SRRL. mid_dim is the average of in_dim and out_dim.

Projector Design In CRD (Tian et al., 2020), the projection module to $\mathbb{R}^d$ is a linear matrix multiplication. Since its publication, improvements on the contrastive learning regime have been proposed, one of which is the use of a *deeper* projector (Chen et al., 2020b;a). Thus, we also compared using the 2-layer MLP architecture found in MoCoV2 (Chen et al., 2020b) vs. a linear one. The exact design can be found in Figure 7. For our A/U based loss, we found that both the deeper and linear one achieved good performance but chose the deeper one to show the flexibility of our design. Our source code provides different options for the choice of the projector. Substituting the deeper projector in CRD exactly describes CRD++.

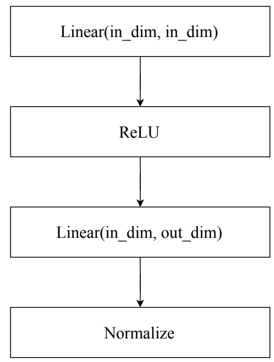


Figure 7: The design of the projector $g(\cdot)$ in A/U and CRD++.

A.5 EXPERIMENT DETAILS

For training, linear probing, and distilling, we used the AdamW optimizer (Loshchilov & Hutter, 2019) with the default parameters: 1e-3 initial learning rate, beta = (0.9, 0.999), epsilon=1e-8 and a weight decay of 0.01. No additional learning rate schedulers were used. All experiments were done for 250 epochs with a random seed of 9. Inputs were centered, resized to 224x224, and randomly flipped horizontally. Pre-trained models were taken from PyTorch's model hub [2].

A.6 DATASET DETAILS

We used the Indoor Scene Recognition dataset (MIT-67) Quattoni & Torralba (2009), the Caltech-UCSD Birds-200-2011 dataset (CUB-2011) Wah et al. (2011), the Describable Textures Dataset (DTD) Cimpoi et al. (2014), and CIFAR-10/100 Krizhevsky et al. (2009). The first 4 are considered data-limited. MIT-67 has 67 classes with 5360 images in the train set. CUB-2011 has 200 classes with 5994 images in its train set. Caltech-101 has 6907 train images across 101 classes, though some classes have many more (hundreds) of images while others have as low as 30. DTD has a train set of size 1880 for 47 classes. Lastly, CIFAR-10 and CIFAR-100 are standard vision datasets with 50,000 training images across 10 and 100 classes, respectively.

[2] https://pytorch.org/vision/stable/models.html