Deep Classifier Mimicry without Data Access

Steven Braun[1] **Martin Mundt**[1,2] **Kristian Kersting**[1,2,3,4]

[1]Department of Computer Science, TU Darmstadt, Darmstadt, Germany
[2]Hessian Center for AI (hessian.AI), Darmstadt, Germany
[3]German Research Center for Artificial Intelligence (DFKI), Darmstadt, Germany
[4]Centre for Cognitive Science, TU Darmstadt, Darmstadt, Germany
{steven.braun,martin.mundt,kersting}@cs.tu-darmstadt.de

Abstract

Access to pre-trained models has recently emerged as a standard across numerous machine learning domains. Unfortunately, access to the original data the models were trained on may not equally be granted. This makes it tremendously challenging to fine-tune, compress models, adapt continually, or to do any other type of data-driven update. We posit that original data access may however not be required. Specifically, we propose Contrastive Abductive Knowledge Extraction (CAKE), a model-agnostic knowledge distillation procedure that mimics deep classifiers without access to the original data. To this end, CAKE generates pairs of noisy synthetic samples and diffuses them contrastively toward a model's decision boundary. We empirically corroborate CAKE's effectiveness using several benchmark datasets and various architectural choices, paving the way for broad application.

1 INTRODUCTION

In the contemporary machine learning landscape, the rise in availability of pre-trained models has significantly facilitated development of downstream applications. In conjunction with prominent underlying techniques, ranging from parameter pruning and sharing (Han et al., 2015; Wang et al., 2016), low-rank factorization (Yu et al., 2017; Denton et al., 2014), to knowledge distillation (Hinton et al., 2015), these pre-trained models can now be efficiently fine-tuned, compressed, or even adapted continually. Enabling all the latter through a single mechanism, knowledge distillation seems to be a particularly promising contender from the plethora of available options. At its core, it aims to transfer the knowledge from a (typically larger, more complex) teacher model to a (typically smaller, simpler) student model by training the student to mimic the teacher's predictions, feature responses, or other inferrable quantities from the learned function. Such mimicry then enables the student to reach similar performance levels, at reduced computational cost and memory usage or allow a model to continue learning, if the student is the same model that retains knowledge from a prior time step (Li and Hoiem, 2016). However, the knowledge distillation optimization procedure traditionally requires access to original data. Unfortunately, a provided model state may not be accompanied with all its training data or access to the latter may deliberately not be granted.

Despite an impressive amount of ensuing applications in natural language processing (Tang et al., 2019; Mou et al., 2016; Jiao et al., 2020), computer vision (Liu et al., 2017; Zhou et al., 2018; Yim et al., 2017), and speech recognition (Hinton et al., 2015; Lu et al., 2017; Ramsay et al., 2019), the majority of approaches is thus still limited by a host of assumptions. In most works, students train on the original training data (Hinton et al., 2015; Chen et al., 2019; Han et al., 2021) or additional generative auxiliary models are used to approximate the data distribution (Chen et al., 2019; Han et al., 2021). Alternatively, the necessity of data can be alleviated by imposing heavy constraints on architectures (Chen et al., 2019; Han et al., 2021; Yin et al., 2020; Nayak et al., 2019). These dependencies limit the applicability of knowledge distillation when the original training data is not available, the teacher and student model architectures mismatch,

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

or training additional generative models is infeasible. However, a crucial realization is that a majority of these tasks may not even require strong assumptions if one accounts for the task's nature.

Specifically, *we posit that supervised scenarios and classification, in particular, do not require all knowledge to be distilled*. On the contrary, it is *the decision boundary that needs to be closely mimicked* by a student. We refer to respective distillation as **abductive knowledge extraction**.

Based on this realization, we lift prior works' assumptions and propose the first knowledge distillation method that is truly model-agnostic, i.e. operating effectively without reliance on any specific architectural features, components, or configurations of the teacher (or the student) model, while not requiring access to any original training data. To this end, our introduced Contrastive Abductive Knowledge Extraction (CAKE) generates synthetic data pairs via a contrastive diffusion process, which are directed toward opposing sides of a teacher model's decision boundary. In symbiosis, a contrastive pull ensures that a prospective student trains on samples that closely support the teacher's decision, whereas induced noise scatters samples to sweep relevant regions along the boundary. Fig. 1 shows an intuitive "two-moons" example, where CAKE is compared to naive synthetic samples based on gradient descent alone and a generative model. As detailed later, CAKE succeeds where competitors fail at data-free model-agnostic knowledge distillation due to collapse to trivial solutions or failure to cover a broad spectrum close enough to the relevant decision boundary.

- We introduce Contrastive Abductive Knowledge Extraction (CAKE), a model-agnostic knowledge distillation procedure without access to original data. Instead, a contrastive diffusion process generates synthetic samples that border a teacher's decision boundary.
- We empirically highlight the contribution of CAKE's components, showcase how teacher and student neural networks can differ in depth and capacity, and analyze CAKE's effectiveness when teacher and student models differ (MLP, CNN, ResNet, and ViT).
- We corroborate that CAKE's classification accuracy is competitive with a variety of "state-of-the-art" methods that require data access or heavy model assumptions.

We provide open-source code for CAKE at `https://github.com/ml-research/CAKE`.

2 KNOWLEDGE DISTILLATION AND THE CHALLENGE OF DATA AVAILABILITY

In this section, we will discuss the key concepts behind knowledge distillation, briefly explore the different types of distilled knowledge and distillation schemes, and summarize limitations with respect to data availability commonly found in the literature and respective surveys (Gou et al., 2021; Liu et al., 2021).

2.1 Distillation in Supervised Classification

The original variant of knowledge distillation introduced by Hinton et al. (2015) uses a softened version of the teacher's (logit) output to train a student model to mimic the teacher. At the example of supervised classification, given a training dataset with N input-target pairs $(\boldsymbol{x}_i, y_i)$, a student f^{S}, and a teacher f^{T}, we denote $\boldsymbol{z}_i^{\mathrm{S}} = f^{\mathrm{S}}(\boldsymbol{x}_i)$ and $\boldsymbol{z}_i^{\mathrm{T}} = f^{\mathrm{T}}(\boldsymbol{x}_i)$ as the student and teacher logits respectively. The student is trained by minimizing a loss function $\mathcal{L}$ that balances the prediction of ground truth labels and matches the softened output of the teacher $p(\boldsymbol{z}_i, \tau) = \left(\exp\left(z_i^1/\tau\right)/Z_i, \ldots, \exp\left(z_i^C/\tau\right)/Z_i\right)$, where $Z_i = \sum_j \exp(z_i^j/\tau)$ is the normalization constant and τ is a temperature parameter that controls the softness of the output distribution. The full student training objective thus becomes a conjunction of true labels and the teacher's "soft labels":

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{x}_i, y_i\right) = \lambda_1 \mathrm{CE}\big(y_i, p\big(\boldsymbol{z}_i^{\mathrm{S}}, 1\big)\big) \\ + \lambda_2 \mathrm{CE}\big(p\big(\boldsymbol{z}_i^{\mathrm{T}}, \tau\big), p\big(\boldsymbol{z}_i^{\mathrm{S}}, \tau\big)\big) \quad , \end{aligned} \quad (1)$$

where λ_1 and λ_2 are hyperparameters that control the trade-off between the two terms and CE is the cross-entropy loss. The first term ($\mathcal{L}_{\mathrm{NLL}}$) in the loss function encourages the student to predict the ground truth labels, while the second term ($\mathcal{L}_{\mathrm{KD}}$) tries to match the softened output of the teacher.

More generally, knowledge distillation techniques can be categorized based on the *distilled knowledge* and *distillation schemes*. Distilled knowledge may include response-based methods focusing on model outputs (Hinton et al., 2015), feature-based methods targeting intermediate representations (Zagoruyko and Komodakis, 2017; Chen et al., 2017), and relation-based methods capturing pairwise relations between instances (Yim et al., 2017; You et al., 2017). Distillation schemes may encompass offline distillation, which involves pre-trained teacher models, online distillation where teacher and student models are trained simultaneously, and self-distillation, where the teacher and student are the same model (Zhang et al., 2019; Hou et al., 2019; Yang et al., 2019). The choice of

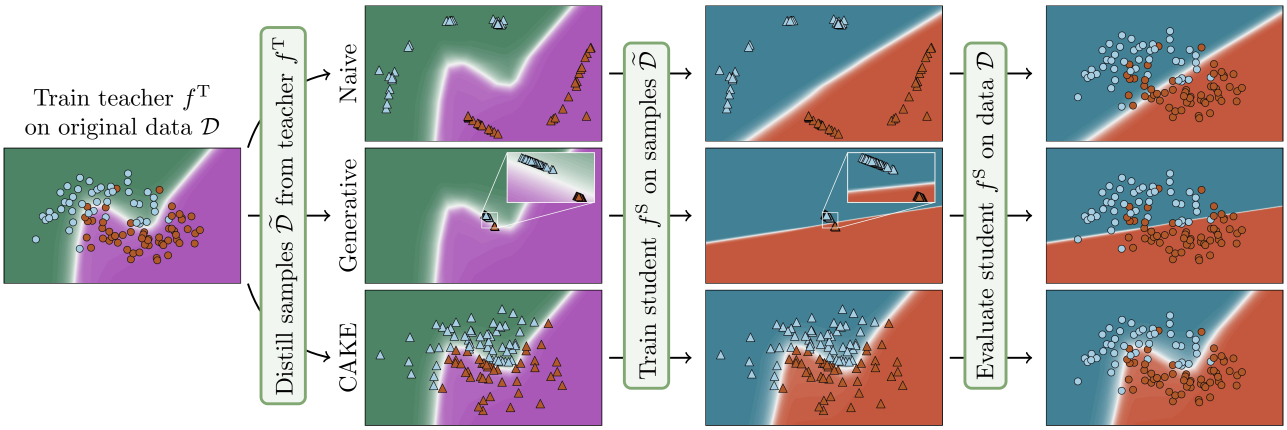

Figure 1: Comparison of naive, generative, and CAKE methods for knowledge distillation on the two-moons dataset. The background visualizes teacher (green/purple) and student (blue/red) decision functions, juxtaposed with original data (○) and synthesized samples (△). Naive and generative methods often converge to similar local minima, inducing an ineffective student decision function. In contrast, CAKE generates samples across the entire decision-relevant region, resulting in a student model that accurately learns the data decision function if trained exclusively on its synthetic samples.

the distillation scheme depends on an application's requirements, including computational resources, training data, and desired accuracy.

2.2 Distillation without Data Access

While the seminal work of Hinton et al. (2015) introduced knowledge distillation with the student having access to the original training data and using the smoothed teacher outputs as *additional information*, knowledge distillation can be further lifted to a *"data-free"* setting. Here, data-free refers to providing no access to the data distribution that the teacher was trained on. The focus is then to construct synthetic samples from the teacher that serve as exclusive training data for the student. There are generally two approaches to achieve such generation of synthetic data.

One angle makes use of generative models to synthesize samples that are relevant to the teacher's objective, therefore extracting knowledge into an auxiliary generative model that learns to sample the data distribution. Recent examples are adversarial distillation, such as DAFL (Chen et al., 2019) and its successor RDSKD (Han et al., 2021) which employ a generative adversarial network (GAN) Goodfellow et al. (2014). However, while students are trained with synthetic GAN samples, the training procedure of the GAN itself again requires access to original data to construct the adversarial objective. Further, an additional model now needs to be carefully crafted, which may be prone to issues such as e.g. mode collapse in GANs.

The alternative angle is to leverage the teacher's parameters directly to construct a synthetic dataset. The initial work, DeepDream (Mordvintsev et al., 2015), uses an image prior and performs gradient descent on an input image w.r.t. maximizing a specific class probability. The later DeepInversion (Yin et al., 2020) uses DeepDream's total variation and the l_2-norm as an image prior and extends the optimization objective by a feature distribution regularization term. This imperative term measures the l_2-distance between convolution activations and respective BatchNorm (Ioffe and Szegedy, 2015) statistics, as the latter provides a simple Gaussian proxy to the encoded distribution. Wang (2021a) extends this principle by modeling the intermediate feature space with a multivariate normal distribution. However, methods such as DeepInversion then entail a restriction on the teacher requiring specific layers. As such, the approach cannot be applied if the teacher is either treated as a black box with no access to intermediate outputs or does not contain the necessary functions. CAKE follows in these works' footsteps, but lifts the constraints on model architecture and intermediate value access.

2.3 The Pitfalls when Removing Data Access without Model Constraints

To contextualize prior works and highlight the challenge of removing both access to data and intermediate values of specific model functions, we circle back to the earlier shown figure Fig. 1. On the basis of the simple 2-D two-moons example, the top row depicts original (circles) and synthetic (triangles) data for the naive DeepDream approach. As the latter optimizes initially random inputs solely to maximize the cross-entropy loss, the first common pitfall ensues. Namely, *samples easily satisfy maximum confidence if they lie far away from the decision boundary.* When a student is trained on these samples, the decision boundary is overly simplistic, here leading to a linear decision that is incorrect for the original task. The second row shows a respective generative model trained to synthesize data that minimize the teacher's confidence. Whereas the first pitfall may also occur, we can condition samples and contrast pairs (as in the later CAKE for direct comparison). However, a second caveat now arises. Namely, *parameterized generators may easily collapse towards trivial solutions or sample select regions.* As they collapse to specific modes that do not cover the distribution necessarily, the student's solution may once more be inadequate for the original task.

3 CAKE: CONTRASTIVE ABDUCTIVE KNOWLEDGE EXTRACTION

In the previous section, we expounded on the limitations and assumptions associated with existing knowledge distillation techniques when original training data is unavailable and strict model assumptions cannot be made. To overcome these challenges, we now introduce Contrastive Abductive Knowledge Extraction, CAKE for short. In contrast to prior works, CAKE extracts the abductive knowledge of a teacher in a fully model-agnostic way that does not require any original data.

3.1 Contrasting the Decision Boundary: Abductive Knowledge Extraction

We propose a conceptual shift in the objective of the distillation procedure. Contrary to the emphasis placed by a significant portion of the knowledge distillation literature on the visual fidelity and closeness to original data, we argue that the ultimate goal is not to accurately emulate the data-generating distribution. Instead, it should be to sample effectively along the decision boundary region, such that a student can later mimic it. With this in mind, we propose to create pairs of noisy synthetic samples and employ a contrastive approach to diffuse them towards the decision boundary. Intuitively, think of drawing two samples for two different classes (or sets in multi-class scenarios) and pulling both towards each other until their predicted label gets swapped. To this end, we employ the squared Euclidean distance between logit pairs for synthetic samples of different classes:

$$\mathcal{L}_{\mathrm{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\left[y_i \neq y_j\right] \left\| f^{\mathrm{T}}(\boldsymbol{x}_i) - f^{\mathrm{T}}(\boldsymbol{x}_j) \right\|_2^2 \quad . \tag{2}$$

Note that despite the availability of elaborate contrastive formulations (van den Oord et al., 2019; Song et al., 2016; Frosst et al., 2019), we focus on initial simplicity. To avert the risk of synthetic samples collapsing into a single region that minimizes the objective, recall the second row of Fig. 1, it becomes necessary to further disperse these samples along the decision boundary, as we elaborate in the following subsection.

3.2 Sweeping the Decision Boundary: Implicit and Explicit Noise

Having developed an objective aimed at generating samples close to the decision boundary, we must acknowledge that this objective does not yet ensure extensive coverage *along* the decision boundary. On the one hand, abductive knowledge extraction need not perfectly reflect the data distribution. However, on the other hand, it is imperative to mimic a wide range of the decision boundary. We therefore require an additional mechanism to explore along the decision boundary. We posit that such exploration can be achieved through the introduction of noise into the sample update. As the contrastive term already acts as a perpendicular force, ensuring closeness between sets of samples between classes, the injection of noise effectively diffuses them in parallel to the decision boundary. In CAKE, we thus inject noise by means of the well-understood stochasticity of SGD-based optimizations and common step size schedules. Again for initial simplicity, we choose a simple linear schedule, but we note that a plethora of variants for noisy estimates exist. This effectively causes the optimization to disperse the synthetic samples along the decision boundary.

While CAKE presents an intuitive, highly empirically effective, but perhaps somewhat ad-hoc, solution to the induction of noise, we now also propose a more principled formulation. Recent advances in generative modeling have rediscovered the importance of noise through the integration of diffusion processes. Following this spirit, we introduce a CAKE variant termed Langevin Abductive Knowledge Extraction (LAKE). In the latter, we incorporate noise into the synthesis procedure with Langevin dynamics based diffusion, generating samples from noisy gradients of the input:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \mathcal{L}\big(\boldsymbol{x}_i^t\big)\,\eta(t)\,\nabla_{\boldsymbol{x}}\mathcal{L}\big(\boldsymbol{x}_i^t\big) + \sqrt{2\varepsilon_i^t\eta(t)}\boldsymbol{\varepsilon}_i^t \quad , \tag{3}$$

with $\boldsymbol{\varepsilon}_i^t \sim \mathcal{N}(0, \mathbf{I})$ for $t = 1, \ldots, T$. The process will converge samples according to the true distribution defined by the loss landscape, as both $T \to \infty$ and $\eta(t) \to 0$. The diffusion property of the Langevin update step aids in dispersing samples along the decision boundary, thus preventing collapse. However, the theoretical guarantees only hold for the limit $T \to \infty$ and

$\eta(t) \rightarrow 0$ and further empirical findings seem to indicate that the explicit Gaussian noise term in the diffusion process may not be fully necessary (Bansal et al., 2022; Daras et al., 2022). Ultimately, we emphasize that the presence of noise seems to be crucial, as also highlighted by the quantitative results for CAKE and LAKE in subsequent Section 4, but the choice w.r.t a potential trade-off between empirical results and rigor is left to the prospective user.

3.3 Injecting Auxiliary Domain-specific Knowledge: The Role of Data Priors

In addition to our rigorous premise of no access to original training data, we acknowledge that information about the data domain typically exists. That is, even when a pre-trained model contains no reference to real data, its purpose and domain of application is typically made obvious. There is no conflict in integrating such auxiliary knowledge into the synthesis process through data priors. For instance, when the application is image-based, we can employ a total-variation prior, as initially used also by DeepDream for the purpose of "generating beautiful art" from random noise:

$$\mathcal{L}_{\text{TV}}(\boldsymbol{x}) = \sum_{i=1}^{H} \sum_{j=1}^{W} \|\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i-1,j}\| + \|\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i,j-1}\| \ . \tag{4}$$

Here, $\boldsymbol{x}$ represents an image of dimensions $H \times W$, and $\boldsymbol{x}_{i,j}$ corresponds to the pixel at the location (i, j). Intuitively, this prior mirrors our expectation that inputs are images, and we thus expect depicted concepts to be locally consistent. More generally, such priors enable injection of potential meta-knowledge we may possess in the form of constraints that facilitate the synthetic sample optimization. Whereas our work later showcases popular image classification, imagine e.g. a prior on the range of expected numerical values when confronted with tabular data as a second example.

3.4 The Overall CAKE Algorithm

For completeness, we lay out the full CAKE procedure in Appendix A. Conceptually, all synthetic samples could be generated in parallel. However, due to both practical compute and memory constraints, and to make the injection of noise more intuitive, the algorithm outlines the generation of M sets of synthetic sample "mini-batches". For each mini-batch $\widetilde{\mathcal{D}}_m$, N random synthetic samples and labels $(\widetilde{\boldsymbol{x}}_i^{t=0}, \widetilde{y}_i)$, where $\widetilde{\boldsymbol{x}}_i^{t=0}$ and $\widetilde{y}_i$ are drawn from priors of our choice $p(\boldsymbol{x})\,, p(y)$. Subsequently, the algorithm iterates over the number of synthetic samples per mini-batch, $\frac{N}{M}$, and for each sample, it performs T iterations. Within each iteration, the samples $\widetilde{\boldsymbol{x}}_i^t$ are fed through the teacher f^{T} to obtain logits $\boldsymbol{z}^{\mathrm{T}}$. Then, we compute the extraction loss l as a weighted mixture of $\mathcal{L}_{\text{KD}}, \mathcal{L}_{\text{contr}}$, and $\mathcal{L}_{\text{TV}}$. An update step with scheduled step size $\eta(m)$ is performed on the synthetic sample as specified in Section 3.2. The algorithm ultimately returns the union of all synthetic mini-batches, $\widetilde{\mathcal{D}} = \bigcup_{m=1}^{M} \widetilde{\mathcal{D}}_m^T$. We can then proceed and train a student model on the newly synthesized dataset. In CAKE, we argue that the necessary noise can be induced intuitively as a function of the current mini-batch. Respectively, for LAKE we replace line 8 with earlier Eq. (3).

Table 1: Ablation analysis of CAKE & LAKE in distilling a ResNet-34 to a ResNet-18 on CIFAR-10, highlighting that inclusion of each individual component is meaningful to the overall performance of our method.

Setting	Student Accuracy	
	LAKE	CAKE
baseline	28.0 ± 3.16	15.6 ± 4.15
$+\mathcal{L}_{\text{KD}}$	34.8 ± 5.56	19.2 ± 4.91
$+\mathcal{L}_{\text{contr}}$	24.7 ± 2.32	39.7 ± 7.27
$+\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{contr}}$	36.8 ± 3.02	42.5 ± 8.13
$+\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{contr}} + \mathcal{L}_{\text{TV}}$	58.6 ± 4.02	71.0 ± 3.75

4 ABLATION STUDIES ON CIFAR

To highlight the contributions of the design elements introduced in Sections 3.1 to 3.3 and to corroborate their utility beyond the two-dimensional Fig. 1, we start with ablation studies on CIFAR-10. Table 1 shows respectively obtained student accuracies for CAKE and the LAKE variant for a ResNet-34 (He et al., 2016) teacher and smaller ResNet-18 student, both trained for 30 epochs on batches of size 256 with SGD and a learning rate of 0.5 scheduled with OneCycleLR (Smith and Topin, 2018). We extract 500 mini-batches of 256 samples with loss weights $\lambda_{\text{contr}} = 1\text{e}1, \lambda_{\text{cls}} = 1\text{e}3$, and $\lambda_{\text{TV}} = 1\text{e}5$ for 256 iterations and an initial step size of 0.1. Further details follow standard practice and are provided in Appendix A and Appendix B. As described in earlier sections, we introduce noise by linearly decaying the step size across four magnitudes for the sample synthesis in CAKE and through Langevin diffusion in LAKE.

We can observe that the baseline, where synthetic samples are generated solely by maximizing cross-entropy, delivers a student accuracy of only 28.0% for LAKE and 15.6% for CAKE, demonstrating the necessity of additional synthetization terms. The addition of the knowledge distillation loss ($\mathcal{L}_{\text{KD}}$) improves the performance for both LAKE and CAKE, increasing student accuracy to 34.8% and 19.2% respectively, indicating, that the use of teacher-based soft labels is not as effective when applied to synthetic data compared to original training data. However, the contrastive loss $\mathcal{L}_{\text{contr}}$ presents a more nuanced scenario. Whereas it significantly enhances the performance of CAKE to 39.7%, it seems to slightly degrade performance for LAKE's to 24.7% when taken on its own, suggesting that $\mathcal{L}_{\text{contr}}$ is more beneficial in the absence of noise. However, when $\mathcal{L}_{\text{KD}}$ and $\mathcal{L}_{\text{contr}}$ are combined, both LAKE and CAKE exhibit improved performance, with student accuracy reaching 36.8% and 42.5% respectively, illustrating the complementary nature of these two loss components. The final addition of $\mathcal{L}_{\text{TV}}$ as a means to induce prior knowledge about the possible structure of image-based data leads to large improvements in performance of both LAKE and CAKE, resulting in student accuracies of 58.6% and 71.0%.

5 CAKE ACROSS SCALES

Following the spirit of the original knowledge distillation paper's experiments and goals (Hinton et al., 2015), we investigate CAKE's ability for model compression on CIFAR-10. To this end, we employ the well-known family of ResNet models with varying depths as both teachers and students. Specifically, we consider the following depths, with the number of parameters denoted in millions in brackets: 152 (58.2M), 101 (43.5M), 50 (23.5M), 34 (21.3M), 18 (11.2M), and 4 (1.2M). Fig. 2 shows respective groups of bars for achieved test accuracy for each teacher depth on the x-axis, where the hatched bars (with further explicit "teacher" bar label) show the respectively sized teacher's performance. All other bars quantify students of varying depths. For convenience, student depth is provided explicitly as the bar label and an applied shading further highlights smaller models through lighter shading.

From the obtained accuracies, it becomes evident that CAKE displays stable performance across various teacher-student model capacities. Most importantly, even smaller student models (ResNet-18, ResNet-34) exhibit competitive performance when distilled from deeper teachers, indicating the effectiveness of CAKE in compressing knowledge of previously overparametrized models. As such, if the teacher model's complexity decreases too much, here in the case of ResNet-4, the student models also feature a significant drop in accuracy, irrespective of their depth. Naturally, this suggests that the lower capacity of the teacher model limits the quality of knowledge it can provide, which the student cannot recover at any capacity. Not surprisingly, a student model with very limited capacity suffers from bottlenecked information and amplifies performance degradation when knowledge is distilled, resulting in expected inferior performance across all examined teacher model capacities.

6 CAKE ACROSS MODEL TYPES

A key advantage of CAKE does not only lie in its effectiveness without access to original data but also in the fact that there are no imposed constraints on model architectures or required intermediate model values for distillation. Much in contrast to the earlier mentioned prior works that require models to be of the same type or functioning on the premise of batch normalization layers, we are thus free to distill knowledge between a teacher and student model of different types. In fact, our sole requirement is that a model API implements a black box differentiable "forward" and a "backward" pass, where it is sufficient to simply obtain the final input gradient without any in-between states.

In the following, we thus investigate the performance of CAKE between four popular neural network types: 1) *Multi-layer Perceptrons (MLP)*, 2) *Convolutional Neural Networks (CNN)*, 3) *ResNet-4*, and 4) *Vision Transformer (ViT)* (Dosovitskiy et al., 2021). For fair comparison, we have matched the models' parameter amounts, see appendix for details. Fig. 3 shows the result for *across-model type* distillation on MNIST (LeCun et al., 1998a), combining every model type with every other. Each group on the x-axis describes a set of experiments with a specific teacher type, where the teacher's accuracy is hatched. The intra-group bars represent student results when trained on the synthetic samples of the particular teacher.

Overall, we find that distillation from an MLP to any other model is effective across the board, while the distillation performance from a ResNet to other models is generally poor. Importantly, the distillation performance is notably robust when both the teacher and student models share the same model type. Following these results, we first emphasize that our chosen models have all been roughly matched in terms of overall parameter amount and achieve negligibly similar teacher test accuracies. Thus, we hypothesize that when teacher and student models *have similar inductive biases, the distillation process tends to be most effective*. In addition, as observable in the case of MLPs that have less inductive biases than the other contenders, it seems that the majority of students can also excel as they are unrestricted in forming their own auxiliary assumptions. Here, the only exception is the ViT, for which distillation results are mixed and consistently underperform, unless the teacher is also a ViT. We further conjecture that this outcome could be attributed to the fundamentally distinct manner in

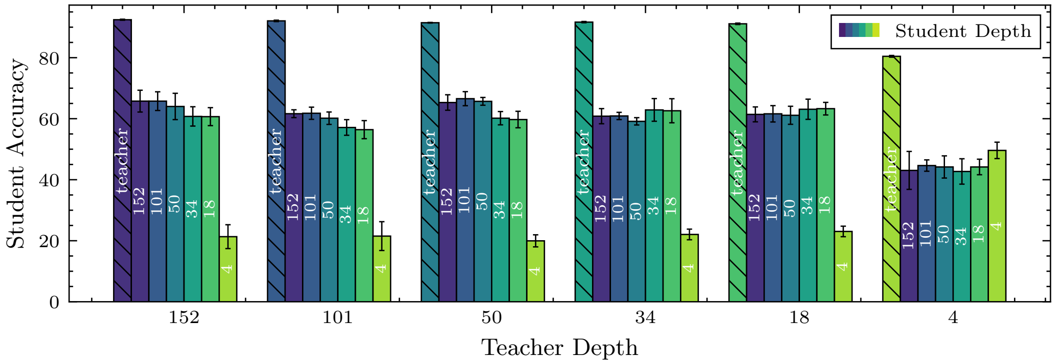

Figure 2: Student model accuracy on CIFAR-10 (y-axis) when trained on synthetic data distilled from ResNet teacher models of different depths. Each group of bars corresponds to a ResNet teacher model of a particular depth (x-axis), and each bar within a group shows the accuracy of the student model distilled from that teacher model, along with its standard deviation as error bars. As desired, CAKE can compress models at a stable accuracy until capacity is too heavily constrained.

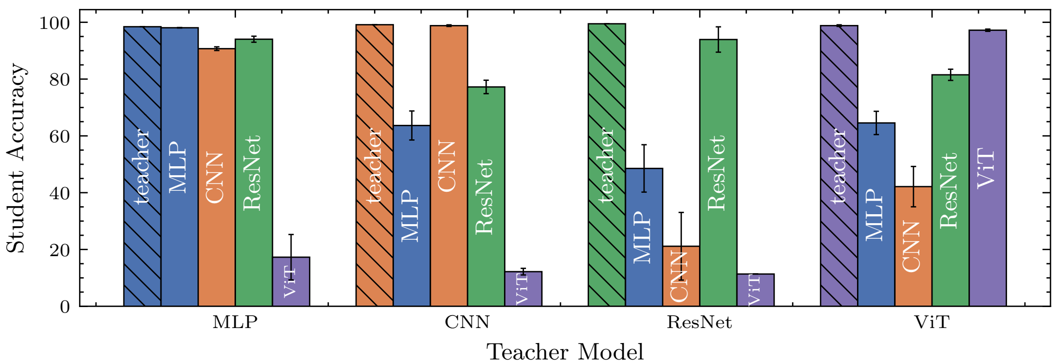

Figure 3: Performance of different student models distilled from teacher models of various model types trained on MNIST: CNNs, MLPs, ResNets, and ViTs (parameter amounts are set to be similar). Each group of bars corresponds to a particular teacher type and each bar within a group shows the accuracy of a particular type of distilled student model, along with its standard deviation as error bars (5 trials). Overall, matching model types consistently provides good results, whereas distillation across types seems to work if the teacher has less inductive bias than the student.

which inputs are fed into the model, specifically, the tokenization into sequences in ViTs. However, most importantly, as we track Fig. 3 to the right, our analysis thus suggests a rather simple rule of thumb. *When in doubt of the teacher's type, choosing a ResNet model appears to be a safe choice*, as it provides stable performance independently of the source model.

7 CAKE VS. TAILORED METHODS

Having evaluated the key factors contributing to CAKE's performance, its efficacy in model compression, and its ability to distill across diverse model types, we now position CAKE within the larger context of existing techniques. As discussed in Section 2, these methods often require access to original data, are tailored to specific models, or impose both conditions. We include a wide set of techniques, their assumptions, and performances on MNIST (LeCun et al., 1998a), FMNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR (Krizhevsky, 2009) in Table 2 (see Appendix D for an extended table). Despite all other techniques imposing strict requirements on model type and data availability, our results show compelling evidence that CAKE can effectively lift assumptions with little to no performance detriment.

For both MNIST (LeNet-5 to LeNet-5-Half) and SVHN (ResNet-34 to ResNet-18) settings, CAKE achieves comparable student accuracy to other techniques, despite the latter requiring data access and/or being model-specific. In the CIFAR-10 scenario (ResNet-34 to ResNet-18), we attain a student accu-

Table 2: Comparison of knowledge distillation techniques, presenting teacher and student model accuracies, to highlight that CAKE is effective despite lifting typical model constraints (MA: model-agnostic) and requiring no data access (DF: data-free). Note that standard deviations are typically not reported in the literature, obfuscating potential volatility in reproduction. See Appendix D for further details on the compared methods.

Method	DF	MA	Dataset	Teacher	Acc.	Student	Acc.
KD	✗	✓	MNIST	LeNet-5	99.3	LeNet-5-Half	98.8
			FMNIST	LeNet-5	90.8	LeNet-5-Half	89.7
			CIFAR-10	ResNet-34	95.6	ResNet-18	94.3
DAFL	✓	✗	MNIST	LeNet-5	97.9	LeNet-5-Half	97.6
			CIFAR-10	ResNet-34	93.7	ResNet18	90.4
DI	✓	✗	CIFAR-10	ResNet-34	95.4	ResNet-18	91.4
ADI	✓	✗	CIFAR-10	ResNet-34	95.4	ResNet-18	93.3
DD	✓	✓	CIFAR-10	ResNet-34	95.4	ResNet-18	30.0
ZSDB3KD	✓	✓	MNIST	LeNet-5	99.3	LeNet-5-Half	96.5
			FMNIST	LeNet-5	91.6	LeNet-5-Half	72.3
			CIFAR-10	AlexNet	79.3	AlexNet-Half	59.5
CAKE	✓	✓	MNIST	LeNet-5	99.3 ± 0.12	LeNet-5-Half	98.4 ± 0.18
			FMNIST	LeNet-5	91.0 ± 0.12	LeNet-5-Half	76.5 ± 1.01
			SVHN	LeNet-5	89.8 ± 0.38	LeNet-5-Half	62.9 ± 4.17
			SVHN	ViT-8	94.4 ± 0.13	ViT-4	83.7 ± 4.77
			SVHN	ResNet-34	96.1 ± 0.08	ResNet-18	94.2 ± 0.54
			CIFAR-10	ViT-8	73.2 ± 0.76	ViT-4	53.8 ± 5.63
			CIFAR-10	ResNet-34	91.8 ± 0.11	ResNet-18	78.9 ± 2.59

racy of 78.9%, almost matching techniques with data access and additional model assumptions with a mere 10%-15% gap. Remarkably, on CIFAR-10, CAKE outperforms DeepDream (30.0% for ResNet-34 to ResNet-18), the only other truly data-free *and* model-agnostic technique, by a factor of two. While ZSDB3KD is data-free and model-agnostic as well, its focus is on model decision outputs and is thus not directly comparable against DD and CAKE.

8 DISCUSSION

We have shown that CAKE can effectively transfer abductive knowledge between models of various capacities as well as models of entirely different types, despite the fact that CAKE lifts previous standard assumptions on models and requirements on original data access. In light of these results, we challenge the current de facto standard of requiring original training data or making model assumptions. Already now, this entails a host of highly interesting future applications of societal significance, as well as an even greater set of prospects once remaining limitations are lifted.

Future Work As CAKE's design lifts the requirement of original data access and simultaneously removes unnecessary model constraints, it now opens up a plethora of future applications and research directions. On the one hand, these lie in performance improvements to our initial intuitive approach. For instance, we can now further make use of the wide array of improved contrastive formulations (van den Oord et al., 2019; Song et al., 2016; Frosst et al., 2019), improvements for diffusion processes, or leverage adaptive signals similar to ADI (Yin et al., 2020) to dynamically steer the distillation process based on the student's performance in the spirit of curriculum learning (Wang et al., 2021). On the other, even more exciting, hand, CAKE's main premise of extracting abductive knowledge also entails that the data distribution is not closely mimicked. This implies that generated synthetic samples do not resemble original data. Fig. 4 showcases synthetic samples generated from a ResNet teacher using CAKE across three datasets: MNIST, SVHN, and CIFAR. A crucial observation from the depicted samples is their lack of visual resemblance to the original training data from the respective datasets. Intuitively, they seem to look more like commonly found adversarial attacks (Ilyas et al., 2019), but note that our synthetic data doesn't serve the same purpose to trick a classifier towards misclassification by perturbing original data. This characteristic might offer a foundational premise for robust privacy-preserving methodologies. In light of these findings, future research could probe the potential of these synthetic samples from a privacy perspective. One avenue to

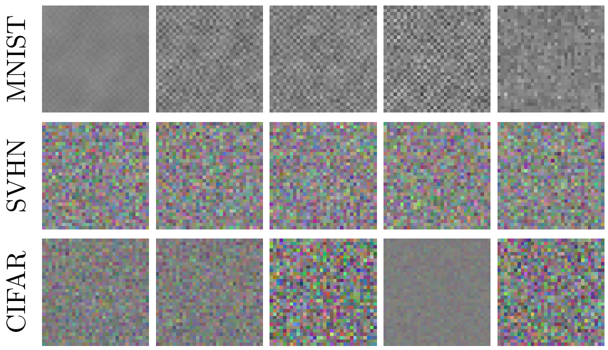

Figure 4: Synthetic samples generated from a ResNet teacher by CAKE on the MNIST, SVHN, and CIFAR datasets, demonstrating no visual resemblance with original training data.

explore is the application of differential privacy, which offer rigorous privacy guarantees by ensuring that the release of statistical information doesn't compromise the privacy of individual data entries. The intersection of synthetic data generation and differential privacy could open up possibilities to ascertain: (1) Whether these synthetic samples meet differential privacy criteria, (2) the potential trade-offs between data utility and privacy when generating such samples, and (3) the robustness of models trained on these synthetic datasets against privacy attacks. In general, CAKE's ability to distill knowledge without data resemblance could be invaluable for applications where data privacy and confidentiality are paramount.

Limitations Although CAKE already yields promising results without common assumptions, we see two remaining limitations to be lifted in the future. First, as highlighted in the previous paragraph, our current distillation process operates independently of the student model. This results in a lack of direct measures to estimate the quality of the synthetic dataset during the distillation phase, potentially limiting the effectiveness of the distillation and the resulting student model's performance. Second, although we assume no access to original data, model constraints, and require no access to intermediate model values, we do nevertheless still require a callable backward function. This does not yet allow CAKE to be used in scenarios where a model is hosted with an API that only allows forward evaluation calls, a limitation we foresee to be overcome through a transition to e.g. the very recent and concurrently proposed forward-forward algorithm (Hinton, 2022).

Societal Impact We raise awareness that model-agnostic abductive knowledge extraction without training data access may be misused to inappropriately extract knowledge from proprietary or confidential models, thereby leading to potential violations of privacy or intellectual property theft. While amicable use cases for e.g. continual or federated learning exist, this also simultaneously highlights the importance of conducting further research into securing our public models. In particular, we foresee that the above-mentioned final limitation of requiring a backward API call may be lifted in the foreseeable future, exposing a crucial issue with current models. Finally, distillation methods will inadvertently mimic existing biases in the teacher model, perpetuating or even exacerbating unfairness or discrimination, potentially making efforts towards data transparency even more challenging.

Acknowledgments

This work was supported by the Federal Ministry of Education and Research (BMBF) Competence Center for AI and Labour ("kompAKI", FKZ 02L19C150) and the project "safeFBDC - Financial Big Data Cluster" (FKZ: 01MK21002K), funded by the German Federal Ministry for Economics Affairs and Energy as part of the GAIA-x initiative. It benefited from the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK; projects "The Third Wave of AI" and "The Adaptive Mind"), and the Hessian research priority programme LOEWE within the project "WhiteBox".

References

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.

Guobin Chen, Wongun Choi, Xiang Yu, Tony X. Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. DAFL: Data-free learning of student networks. In *International Conference on Computer Vision (ICCV)*, 2019.

Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G. Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions. *arXiv preprint arXiv:2209.05442*, 2022.

Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Nicholas Frosst, Nicolas Papernot, and Geoffrey E. Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International Conference on Machine Learning (ICML)*, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision (IJCV)*, 2021.

Pengchao Han, Jihong Park, Shiqiang Wang, and Yejun Liu. Robustness and diversity seeking data-free knowledge distillation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop (NIPS)*, 2015.

Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *International Conference on Computer Vision (ICCV)*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *IEEE*, 1998a.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 1998b.

Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. Data-free knowledge transfer: A survey. *arXiv preprint arXiv:2112.15278*, 2021.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *International Conference on Computer Vision (ICCV)*, 2017.

Liang Lu, Michelle Guo, and Steve Renals. Knowledge distillation for small-footprint highway networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Lili Mou, Ran Jia, Yan Xu, Ge Li, Lu Zhang, and Zhi Jin. Distilling word embeddings: An encoding approach. In *International Conference on Information and Knowledge Management (CIKM)*, 2016.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning (ICML)*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Piyush Raikwar and Deepak Mishra. Discovering and overcoming limitations of noise-engineered data-free knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

David B. Ramsay, Kevin Kilgour, Dominik Roblek, and Matthew Sharifi. Low-dimensional bottleneck features for on-device continuous speech recognition. In *Conference of the International Speech Communication Association (ISCA)*, 2019.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2018.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2019.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. Cnnpack: Packing convolutional neural networks in the frequency domain. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Z. Wang. Data-free knowledge distillation with soft targeted transfer set synthesis. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021a.

Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *International Conference on Machine Learning (ICML)*, 2021b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Chenglin Yang, Lingxi Xie, Chi Su, and Alan L. Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *International Conference on Computer Vision (ICCV)*, 2019.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2018.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]

  (b) Complete proofs of all theoretical results. [Not Applicable]
  (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
  (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
  (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
  (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
  (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
  (a) Citations of the creator If your work uses existing assets. [Yes]
  (b) The license information of the assets, if applicable. [Yes]
  (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
  (d) Information about consent from data providers/curators. [Not Applicable]
  (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
  (a) The full text of instructions given to participants and screenshots. [Not Applicable]
  (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
  (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A THE CAKE ALGORITHM

Algorithm 1 Contrastive Abductive Knowledge Extraction

Require: teacher f^{T}, iterations T, #mini-batches M of N samples, step-size schedule η, priors $p(\boldsymbol{x}), p(y)$

1: **procedure** CAKE($f^{\mathrm{T}}, T, M, N, \eta, p(\boldsymbol{x}), p(y)$)
2: **for** $m = 1$ to M **do** ▷ Number of mini-batches
3: Initialize $\widetilde{\mathcal{D}}_m^{t=0} \leftarrow \left\{ \left(\widetilde{\boldsymbol{x}}_1^{t=0}, \widetilde{y}_1 \right), \dots, \left(\widetilde{\boldsymbol{x}}_N^{t=0}, \widetilde{y}_N \right) \right\}$, where $\widetilde{\boldsymbol{x}}_i^{t=0} \sim p(\boldsymbol{x})$ and $\widetilde{y}_i \sim p(y)$
4: **for** $i = 1$ to N **do** ▷ Number of synthetic samples per mini-batch
5: **for** $t = 1$ to T **do** ▷ Number of iterations
6: $\boldsymbol{z}^{\mathrm{T}} \leftarrow f^{\mathrm{T}}\left(\widetilde{\boldsymbol{x}}_i^t \right)$ ▷ Forward pass through teacher
7: $l \leftarrow \mathcal{L}\left(\widetilde{\boldsymbol{x}}_i^t, \boldsymbol{z}^{\mathrm{T}}, \widetilde{y}_i, \widetilde{\mathcal{D}}_m^t \right)$ ▷ Compute extraction loss
8: $\widetilde{\boldsymbol{x}}_i^{t+1} \leftarrow \widetilde{\boldsymbol{x}}_i^t - \eta(m) \nabla_{\boldsymbol{x}} l$ ▷ Update synthetic samples
9: **return** $\widetilde{\mathcal{D}} = \bigcup_{m=1}^{M} \widetilde{\mathcal{D}}_m^T$

In Algorithm 1 we present the CAKE algorithm. The aim is to extract knowledge from a given teacher model f^{T}. The algorithm requires a teacher model, the number of iterations T, the number of mini-batches M of N samples, a learning rate schedule η, and data and label priors $p(\boldsymbol{x})$ and $p(y)$. The main procedure starts with a loop generating M mini-batches of size N (Line 2). For every mini-batch, synthetic samples are initialized (Line 3). These synthetic samples $(\widetilde{\boldsymbol{x}}_i^{t=0}, \widetilde{y}_i)$ are sampled from the priors with $\widetilde{\boldsymbol{x}}_i^{t=0} \sim p(\boldsymbol{x})$ and $\widetilde{y} \sim p(y)$. Formally, the algorithm then optimizes each of the N sample within the mini-batch (Line 4). Implementation wise, this for-loop can be implemented in a vectorized fashion by parallelizing the computations over the mini-batch axis. For each sample, an inner loop iterates for T iterations (Line 5). In this loop, a forward pass through the teacher model is conducted to compute $\boldsymbol{z}^{\mathrm{T}}$ (Line 6). Following this, the objective loss l is computed, contrasting the synthetic sample with other samples in the batch and using the teacher's output (Line 7). The synthetic sample is then updated based on this loss (Line 8). To implement LAKE as outlined in Section 3.2, this line can be replace with the langevin dynamics update step from Eq. (3). Finally, after processing all samples and mini-batches, we return the joint set of synthetic mini-batches D (Line 9).

B CAKE HYPERPARAMETERS

In our experiments, we defined a set of default hyperparameters for CAKE that we used across all runs unless stated otherwise. This setup involved extracting 500 mini-batches of 256 samples each, with the loss weights $\lambda_{\text{contr}} = 1\text{e}1$, $\lambda_{\text{cls}} = 1\text{e}3$, and $\lambda_{\text{TV}} = 1\text{e}5$ respectively. A mini-batch was generated over 256 update iterations with an initial step size of 0.1 and a linearly decaying step schedule over four magnitudes. As priors in the sample generation process we use a Gaussian distribution $\widetilde{\boldsymbol{x}}_i^{t=0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as data prior $p(\boldsymbol{x})$ and a uniformly distributed categorical distribution over the number of classes $y_i \sim \mathrm{Cat}\left(\frac{1}{C}, \dots, \frac{1}{C}\right)$ as label prior $p(y)$. Furthermore, in the experiments discussed in Section 5, we reduced the number of mini-batches to 250 to reduce the computational load. Conversely, for the experiments in Section 7, the number of mini-batches was increased to 2000 to test the limits of CAKE. One finding from our intermediate experiments was the noticeable improvement in student accuracy as we increased the number of mini-batches. This observation aligns with our intuitive understanding that a higher number of mini-batches allows us to sample the data-relevant decision boundary regions more thoroughly and in a fine-grained manner.

C NETWORK ARCHITECTURES, TRAINING, AND IMPLEMENTATION DETAILS

Architectures. Below we outline the specific network architectures details utilized in our experiments. For each model type, we provide information regarding their architectural configurations, such as the number of layers and their dimensions, as well as the total number of parameters involved. Each of these models has been selected based on its relevance to the data sets used in our experiments, and they represent a variety of model complexities and capabilities. Detailed descriptions for each of these network architectures are provided below.

- **Multi-layer Perceptron (MLP):** An architecture consisting of four hidden layers, each having a hidden size of 100, amounting to a total of 118K parameters.
- **Convolutional Neural Network (CNN):** The architecture of this model featured four convolutional layers. The number of channels for each layer was 32, 64, 64, and 64, respectively, with corresponding kernel sizes of 3, 4, 3, and 3. We employed ReLU activation after each convolution operation, and max pooling was introduced after the second, third, and fourth convolutions. The architecture concluded with a fully connected layer, totaling 107K parameters.
- **LeNet-5 & LeNet-5-Half:** The LeNet-5 model adhered to the architecture proposed by LeCun et al. (1998b), possessing 61.7K parameters. In contrast, the LeNet-5-Half model was a modification of LeNet-5, with the number of filters in each convolution layer reduced by half, resulting in a compact model with 15.7K parameters.
- **Residual Network (ResNet):** The ResNet architectures with depths of 152, 101, 50, 34, and 18 were utilized as described in He et al. (2016). Furthermore, a modified ResNet with a depth of 4 was introduced, where the repetition factor of the layers `conv2` to `conv5` was set to 1. The number of filters in these convolution layers was configured based on the dataset, with 16, 32, 32, and 64 filters for MNIST and 32, 64, 128, and 256 filters for CIFAR-10. The ResNet models accounted for 58M, 43M, 24M, 21M, 11M, 1.2M, and 98K parameters correspondingly.
- **Vision Transformer (ViT):** We employed the ViT architecture for small datasets as suggested by Lee et al. (2021). We had models with depths of 8 (ViT-8) and 4 (ViT-4) with 8 heads, intermediate and MLP dimensions of 64, and a patch size of 4, resulting in 1.1M and 580K parameters respectively. For MNIST experiments, we further reduced model capacity by using a model with a depth of 3 (ViT-3), 4 heads, and intermediate and MLP dimensions of 32, resulting in a compact model with 110K parameters.

Training. In our experiments, both the teacher and student models were trained using the following configurations. The training process was carried out over 30 epochs, utilizing a batch size of 256 for each iteration. The models were optimized using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.5 and weight decay of $1e{-}4$. This learning rate was scheduled using the OneCycleLr Smith and Topin (2018) learning rate scheduler with an initial division factor of 25 and a final division factor of 1e4. For each experiment result we report the mean and standard deviation over five runs with seeds $\{0, \ldots, 4\}$.

Software and Hardware. All of our experiments were implemented in PyTorch (v2.0.0) for model implementation and autograd and the Lightning framework (v2.0.0) for structuring the training pipeline and facilitating model evaluation (see `requirements.txt` for a full list of dependencies and their versions). Experiments were run on A100 GPUs utilizing `bfloat16` precision. The sample generation procedure on a ResNet-34 with a mini-batch size of 250 for 500 batches and 256 update steps per mini-batch takes 40 minutes with an average GPU utilization of 96% occupying 3.725GB of VRAM. All of our code is provided in the supplementary material for full reproducibility and insights.

D DETAILS ON COMPARED METHODS

In this section, we will provide further information on the in Section 7 presented comparison against other relevant KD methods. We further extend Table 2 with more related work in Table 3. For each compared methods, we will reason on the distinctions we made in Table 2, in particular their attributes of being data-free and model-agnostic.

- **Knowledge Distillation** (KD) (Hinton et al., 2015): The seminal work introduced concept of extracting “soft targets” from a teacher model to train a student model. The results from Hinton et al. (2015) were obtained with access to the original training data to fine-tune the student model, thereby inheriting the data-dependency from its teacher model. Hence, it is not data-free. KD has no further assumptions on the teacher or student model architecture making it model-agnostic.
- **Data-Free Learning of Student Networks** (DAFL) (Chen et al., 2019): Utilizes Generative Adversarial Networks (GANs) to train a compact student model without requiring access to the original training data.

The teacher network acts as a fixed discriminator, and a generator produces synthetic samples for student training, making the method data-free. However, DAFL incorporates an architecture-specific activation loss term $\mathcal{L}_a = -\frac{1}{n}\sum_i ||f_T^i||_1$ to guide the learning. This loss term is designed to extract intrinsic patterns from the teacher model's convolutional filters. Thus, DAFL's method is not model-agnostic as it is (a) utilizes a GAN to synthesize samples and (b) closely ties the GAN samples to the architecture of the teacher model due to the architecture-specific activation loss.

- **Robustness and Diversity Seeking Data-free Knowledge Distillation** (RDSKD) (Han et al., 2021): This method addresses the limitations of existing data-free KD techniques by incorporating a specialized loss function that focuses on sample authenticity, class diversity, and inter-sample diversity. It mitigates the conflicting goals of high sample authenticity and class diversity by exponentially penalizing loss increments. Similar to DAFL, RDSK trains a generator to synthesize samples

- **Zero-Shot Knowledge Distillation** (ZSKD) (Nayak et al., 2019): This approach offers a data-free method for knowledge distillation, bypassing the need for original training data or even meta-data. It synthesizes "Data Impressions" from the teacher model, using them as surrogate data for transferring learning to the student model. ZSKD constructs a class similarity matrix and therefore constrains the model to have a fully-connected ultimate layer with a soft-max non-linearity making it model-specific.

- **Gaussian Distillation** (GD) (Raikwar and Mishra, 2022): This technique employs samples drawn from a Gaussian distribution for data-free knowledge distillation. Unlike other attempts at using Gaussian noise, this method provides a reliable solution by addressing the shift in the distribution of hidden layer activations. GD addresses the challenge of "covariate shift" in hidden layer activations through a modification in BatchNorm layers, making the method depend on the presence of such layers in the teacher model.

- **DeepInversion** (DI) (Yin et al., 2020): Generates class-conditional input images by inverting a trained teacher network, making it data-free. However, the method uses a feature distribution regularization term $\mathcal{R}_{\text{feature}}$ that depends on BatchNorm layers in the teacher model to be present, rendering it not model-agnostic.

- **Adversarial DeepInversion** (ADI) (Yin et al., 2020): An extension of DeepInversion. ADI additionally maximizes the Jensen-Shannon divergence between teacher and student network logits to improve synthesized image diversity. Having the same constraints as DI, the adversarial extension now directly creates a dependency between the teacher and student network.

- **DeepDream** (DD) (Mordvintsev et al., 2015): Originating as a blog post, DeepDream serves as a technique for visualizing the learned features of neural networks. It imposes no requirements on the original training data or the architecture of the teacher model, making it both data-free and model-agnostic. DeepDream creates enhanced images by iteratively activating neurons in different layers of a pre-trained network, giving us insights into what the network might "see" or "imagine." This method is thus a fair comparator to CAKE, as it aligns well with our criteria of being data-free and model-agnostic.

- **Zero-Shot Knowledge Distillation from a Decision-Based Black-Box Model** (ZSDB3KD) (Wang, 2021b): Focuses on knowledge distillation when the teacher model is a decision-based black-box, providing only hard labels (class predictions). To overcome this, ZSDB3KD uses the concept of "sample robustness" — a measure of how far a sample lies from the teacher's decision boundaries. With accessible training data, sample robustness helps generate soft labels. When training data is absent (zero-shot), ZSDB3KD creates robust pseudo-samples mimicking the training data distribution. In both cases, knowledge distillation proceeds using generated soft labels. Importantly, ZSDB3KD requires neither the original training data nor specific details of the teacher's architecture, making it data-free and model-agnostic.

Table 3: Comparison of knowledge distillation techniques, presenting teacher and student model accuracies, to highlight that CAKE is effective despite lifting typical model constraints (MA: model-agnostic) and requiring no data access (DF: data-free). Note that standard deviations are typically not reported in the literature, obfuscating potential volatility in reproduction.

Method	DF	MA	Dataset	Teacher	Acc.	Student	Acc.
KD	✗	✓	MNIST	LeNet-5	99.3	LeNet-5-Half	98.8
			FMNIST	LeNet-5	90.8	LeNet-5-Half	89.7
			CIFAR-10	ResNet-34	95.6	ResNet-18	94.3
DAFL	✓	✗	MNIST	LeNet-5	97.9	LeNet-5-Half	97.6
			MNIST	HintonNet	98.4	HintonNet-Half	97.9
			SVHN	WResNet-40-2	95.9	WResNet-16-1	94.3
			CIFAR-10	ResNet-34	93.7	ResNet18	90.4
RDSKD	✓	✗	MNIST	LeNet-5	97.9	LeNet-5-Half	97.6
			SVHN	WResNet-40-2	95.9	WResNet-16-1	94.6
			CIFAR-10	ResNet-34	93.7	ResNet18	90.8
DI	✓	✗	CIFAR-10	VGG-11	92.3	VGG-11	84.2
			CIFAR-10	VGG-11	92.3	ResNet-18	83.8
			CIFAR-10	ResNet-34	95.4	ResNet-18	91.4
ADI	✓	✗	CIFAR-10	VGG-11	92.3	VGG-11	90.8
			CIFAR-10	VGG-11	92.3	ResNet-18	90.7
			CIFAR-10	ResNet-34	95.4	ResNet-18	93.3
ZSKD	✓	✗	MNIST	LeNet-5	97.9	LeNet-5-Half	92.1
			FMNIST	LeNet-5	90.8	LeNet-5-Half	79.6
			SVHN	WResNet-40-2	96.0	WResNet-16-1	14.5
			CIFAR-10	AlexNet	83.0	AlexNet-Half	69.6
			CIFAR-10	ResNet-34	93.7	ResNet-18	10.5
GD	✓	✗	SVHN	ResNet-18	94.5	MobileNetV2	92.9
			CIFAR-10	ResNet-34	93.3	ResNet-18	86.0 ± 0.12
			CIFAR-10	ResNet-34	93.3	ResNet-34	87.1 ± 0.23
DD	✓	✓	CIFAR-10	VGG-11	92.3	VGG-11	36.6
			CIFAR-10	VGG-11	92.3	ResNet-18	39.7
			CIFAR-10	ResNet-34	95.4	ResNet-18	30.0
ZSDB3KD	✓	✓	MNIST	LeNet-5	99.3	LeNet-5-Half	96.5
			FMNIST	LeNet-5	91.6	LeNet-5-Half	72.3
			CIFAR-10	AlexNet	79.3	AlexNet-Half	59.5
CAKE	✓	✓	MNIST	LeNet-5	99.3 ± 0.12	LeNet-5-Half	98.4 ± 0.18
			FMNIST	LeNet-5	91.0 ± 0.12	LeNet-5-Half	76.4 ± 1.01
			SVHN	LeNet-5	89.8 ± 0.38	LeNet-5-Half	62.9 ± 4.17
			SVHN	ViT-8	94.4 ± 0.13	ViT-4	83.7 ± 4.77
			SVHN	ResNet-34	96.1 ± 0.08	ResNet-18	94.2 ± 0.54
			CIFAR-10	ViT-8	73.2 ± 0.76	ViT-4	53.8 ± 5.63
			CIFAR-10	ResNet-34	91.8 ± 0.11	ResNet-18	78.9 ± 2.59