Abstract

Knowledge distillation is a commonly-used compression method in ML due to the popularity of increasingly large-scale models, but it is clear that not all the information a teacher model contains is distilled into the smaller student model. We aim to formalize the concept of “knowledge” to investigate how knowledge is transferred during distillation, focusing on shared invariant outputs to counterfactual changes of dataset latent variables (we call these latents mechanisms). We define a student model to be a good stand-in model for a teacher if it shares the teacher’s learned mechanisms, and find that Jacobian matching and contrastive representation learning are viable methods by which to train such models. While these methods do not result in perfect transfer of mechanisms, we show they often improve student fidelity or mitigate simplicity bias (as measured by the teacher-to-student KL divergence and accuracy on various out-of-distribution test datasets), especially on datasets with spurious statistical correlations.

1 Introduction

Increasingly large deep neural networks (DNNs) trained on huge, web-crawled datasets have shown unprecedented performance on a multitude of tasks [1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25], including emergent capabilities that help with more general-purpose and flexible behaviour with SOTA on tasks they were not finetuned on [26][11][18]. However, resource constraints faced in realistic scenarios, e.g., latency or energy budgets, impact the feasibility of practically deploying such large models. Knowledge distillation [27][28][29][30][31][32][33][34] was motivated as a framework to address this challenge, wherein a smaller “student” model is trained to mimic the outputs produced by the pre-trained “teacher” network on some available dataset. The underlying hypothesis is that enforcing consistency between the outputs produced by two models will yield a “transfer of knowledge” [27], resulting in the less performant model (student) inheriting the mechanisms [35] used by the more performant model (teacher) to make its predictions.

The immense success of distillation in several diverse domains [28][29][30][31][32][33][34][35][36][37][38] does make the argument above sound intuitively correct. However, follow-up work focused on developing a better understanding of knowledge distillation has raised doubt on this viewpoint [39][40][41][42][31][43][44][45]. These works demonstrate that distilled student models infrequently make the same
errors as the teacher models. This is an unlikely result if the models were sharing knowledge (and hence relying on the same mechanisms for making their predictions). These papers make the success of knowledge distillation surprising, and it remains unclear what precise prediction mechanisms, if any, the student inherits from the teacher. Since the student is often observed to outperform the teacher [27, 39, 46, 48, 49], it may be learning entirely novel mechanisms that the teacher does not even possess. This is possible because in practical settings, the distillation dataset is likely of direct relevance to the application of interest. It is also likely smaller than (or minimally overlaps with) the teacher’s pretraining data, and may not be available in offline distillation [27, 48, 49, 50, 51]. Since these smaller distillation datasets are often underspecified (i.e., they contain several predictive attributes that can be used to produce the correct output [52, 53, 54, 55, 56]), a student can in principle learn to match outputs produced by the teacher through a different mechanism to that used by the teacher. Previous explorations into why knowledge distillation works suggests it is unclear what would motivate the student to learn similar prediction mechanisms as the teacher. For example, Cheng et al. [41] suggest that knowledge distillation enforces learning various concepts simultaneously. In addition to providing additional information, they find that teacher outputs guide the optimisation process by preventing excessive exploration of the loss landscape. Phuong et al. [40] find data geometry (e.g., class separation) and optimiser bias to also be contributing factors. This leads to another question: what design decisions in distillation pipelines incentivize the student to learn the same prediction mechanisms as the teacher model? Beyond developing a better understanding of distillation, answering these questions clarify when distillation can be used for producing a student model that serves as a faithful replacement of its teacher counterpart. We make the following contributions:

- **Formalizing knowledge transfer.** We define successful knowledge transfer as when the student and teacher produce the same outputs under systematically generated counterfactuals of a dataset. This definition abstracts away the precise implementation of a prediction mechanism and only emphasizes the behavioral equivalence of two models to define a notion of ‘shared knowledge’.

- **Characterizing knowledge transfer in distillation techniques**. Motivated by our definition, we develop synthetic datasets spanning different modalities to allow counterfactual generation and hence enable precise characterization of which prediction mechanisms a model relies on for producing its outputs. We demonstrate that the standard distillation pipeline of matching teacher logits suffers from a *simplicity bias* [57, 58, 59, 60], resulting in the student learning primarily the simplest mechanisms in the distillation dataset. If the distillation dataset and the teacher’s pretraining dataset have different distributions, the student and teacher may learn entirely distinct prediction mechanisms.

- **Methods for reducing simplicity bias and improving student-teacher matching.** We investigate two distillation methods aiming to more closely match model representations. We find evidence for decreased teacher-to-student KL divergence and less simplicity bias towards certain spurious features.

## 2 Preliminaries: knowledge distillation

**Notation.** Consider a neural network \( f : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^K \) that takes \( n \)-dimensional inputs \( x \in \mathcal{X} \subset \mathbb{R}^n \), has parameters \( \theta \in \mathbb{R}^d \), and produces an output \( f(x; \theta) \in \mathbb{R}^K \) (interpreted as the logits in a classification setting). The neural network predictions are the composition \( f_{\theta} = \text{softmax}_T \circ f \), where \( \text{softmax}_T(z)_i = \frac{\exp z_i/T}{\sum_j \exp z_j/T} \) is the temperature-weighted softmax function for some temperature parameter \( T > 0 \). Cross-entropy loss on a dataset \( \mathcal{D} \in \mathcal{X} \times [K] \) (where \([K]\) denotes the set \( \{1, 2, \ldots, K\} \)) for a model with parameters \( \theta \) is written \( \mathcal{L}(f(\mathcal{D}; \theta)) \).

Let \( \mathcal{D}_t \) be the dataset used to train the teacher model \( f_t : \mathbb{R}^n \times \mathbb{R}^{d_t} \rightarrow \mathbb{R}^k \) (whose parameters are \( \theta_t \in \mathbb{R}^{d_t} \)). The goal in knowledge distillation is to use this teacher model to train a ‘student’ model \( f_s : \mathbb{R}^n \times \mathbb{R}^{d_s} \rightarrow \mathbb{R}^k \) by finding a set of parameters \( \theta_s \in \mathbb{R}^{d_s} \) such that the outputs of the student model \( f_s(\cdot; \theta_s) \) match, in some specific sense, outputs from the teacher model \( f_t(\cdot; \theta_t) \) on a ‘distillation dataset’ \( \mathcal{D}_{\text{distill}} \). We distinguish between the dataset used for training the teacher versus the one used for distilling the teacher into the student to a) emphasize the fact that a practitioner who acquires an off-the-shelf, pretrained teacher model (offline training) is unlikely to have access to the data used for training it, and b) explore situations where the student dataset is markedly different - perhaps more diverse and likely containing spurious mechanisms [35]. There exists a
diverse range of possible distillation methods \cite{28}. Of these, we explore three: Jacobian matching, contrastive representation distillation, and soft targets only from standard distillation. Beyond their widespread use, we choose these methods because they focus only on input/output information – i.e. no intermediate representations are used.

**Standard (base) distillation.** First proposed in the context of neural networks by Hinton et al. \cite{27}, the standard distillation pipeline involves optimizing the agreement between teacher and student model predictions by minimizing the KL-divergence between them:

\[
E_{x \sim D_{distill}} \left[ D_{KL} \left( f_t^{SMT}(x; \theta_t) || f_s^{SMT}(x; \theta_s) \right) \right]
\]

The objective above is equivalent to minimising cross-entropy loss with \( f_t^{SMT}(x; \theta_t) \) as the ‘soft’ targets for a student model. As shown by Hinton et al. \cite{27}, assuming the logits \( f_t(x), f_s(x) \) are zero-centered (mean zero), in the high temperature limit \( T \rightarrow \infty \) this is equivalent to minimising the average logit squared difference: \( \| f_t(x) - f_s(x) \|^2 \).

While a standard cross-entropy loss promoting correct classification is often added to the distillation objective, we follow recent works on understanding knowledge distillation \cite{39,31} and focus on the distillation objective only. No auxiliary classification loss is added while training the student, isolating the distillation objective’s effect in inducing knowledge transfer between teacher and student.

**Jacobian matching.** A Jacobian matching distillation loss matches norm of the gradient of logits with respect to the input between teacher and student: \( f_t^{SMT}, f_s^{SMT} \) and the input-output Jacobians \( \mathbf{J}_{f_t^{SMT}}(\cdot, \theta_t), \mathbf{J}_{f_s^{SMT}}(\cdot, \theta_s) \in \mathbb{R}^h \times \mathbb{R}^h \) match on examples in the distillation dataset \( D_{distill} \). This method is equivalent to classical distillation with analytical addition of perturbation noise to inputs. We can decompose the loss function into one representing usual squared error loss and a regularisation term (Tikhonov regulariser). This does not just match on datapoints, but infinitely many points in their neighbourhood. This can be achieved by adding the following penalty to the standard distillation loss of Eq. [1]

\[
\| \mathbf{J}_{f_t^{SMT}}(\cdot, \theta_t)(x) - \mathbf{J}_{f_s^{SMT}}(\cdot, \theta_s)(x) \|_2^2
\]

Beyond distillation \cite{61,62}, the objective above or variants of it have been introduced in several contexts, such as improving out-of-distribution generalization \cite{63}, improving adversarial robustness \cite{64}, and for learning disentangled representations \cite{65,66}.

**Contrastive distillation.** In contrastive distillation, the goal is to train the student to maximise the mutual information between the representations (typically, the features in the penultimate layer) of the teacher network and the student network on the transfer dataset \( D_{distill} \). In \cite{64}, the authors propose to do this by maximising a lower-bound on the mutual information objective given below. Denote by \( g_t : \mathbb{R}^n \rightarrow \mathbb{R}^h, g_t : \mathbb{R}^n \rightarrow \mathbb{R}^h \) the functions producing the penultimate layer features in the student and teacher models respectively.

**Teacher Representation:** \( Z_t = g_t(X) \)

**Student Representation:** \( Z_s = g_s(X) \)

\[
\mathcal{MI}(Z_t, Z_s) = \mathbb{E}_{p(Z_t, Z_s)}[\log h(Z_t, Z_s)] + N \mathbb{E}_{p(Z_t)p(Z_s)}[\log(1 - h(Z_t, Z_s))]
\]

where \( h : \mathbb{R}^h \times \mathbb{R}^h \rightarrow [0, 1] \) is a learnable function optimized jointly with the parameters of the student; it can be interpreted as an auxiliary “critic” predicting whether the representations were sampled jointly (from \( p(Z_t, Z_s) \)) or independently (from \( p(Z_t)p(Z_s) \)), assuming that they are sampled jointly \( 1/(N + 1) \) of the time. \( h \) typically takes the parametric form:

\[
h(z_t, z_s) = \frac{\exp r_t^T r_s / \tau}{\exp r_t^T r_s / \tau + \exp r_t^T r_s / \tau + \left[ D_{distill} \right]^N},
\]

\[
r_t = W_t z_t / \| W_t z_t \|, \quad r_s = W_s z_s / \| W_s z_s \|
\]

where \( W_t \in \mathbb{R}^{h_{inter} \times h_t}, W_s \in \mathbb{R}^{h_{inter} \times h_t} \) are learnable parameters, and \( \tau, h_{inter} \) are hyperparameters.
Figure 1: Synthetic datasets. Following the protocol above, we embed synthetic cues in two existing datasets: (1) Dominoes [57], where CIFAR-10 images are concatenated with F-MNIST images with the same class number. A third spurious mechanism selects a location of the CIFAR-10 image to set as randomised pixels (we refer to this as ‘box’). If the box mechanism is correlated with CIFAR-10, then the location is determined by the CIFAR-10 label. (2) Spurious parity, where the simple task acts as a spurious mechanism. The label is the parity of the specified hard task subsequence (which is selected for by the control bit). All the hard tasks together act as a single complex mechanism. Shown here is a setup with 3 hard tasks, 8 total hard task bits, 3 hard task bits per task and 2 simple task bits.

3 Defining knowledge transfer

Motivation. As discussed in Sec. 1, despite distillation’s immense success in various fields [28, 29, 36, 30, 31, 37], a formal notion of precisely what knowledge, if any, is transferred from the teacher to student has yet to be defined. For instance, consider a visual object recognition task. In such scenarios, backgrounds are often correlated with the object category due to sampling bias [52, 53, 56]. Here, a model can rely on either the background or more intrinsically meaningful attributes of the object, such as its shape, to solve the recognition task. To understand which, we can evaluate how the model’s prediction changes when image backgrounds are altered. If predictions change, the model relies on information in the (spurious) attribute of image background; if the predictions do not change, the model is invariant to background. Formalizing this intuition, prior work calls use of a predictive attribute to produce outputs a “mechanism” [35], and defines two models that rely on the same mechanisms as mechanistically similar. This framework is relevant to the problem of knowledge transfer in distillation as well. Specifically, if a student model that is perhaps more resource-efficient behaves the same way as a teacher model (i.e. is mechanistically similar), it can serve as a faithful replacement of the teacher. We generalize their formalization to our distillation setup next.

Let $I = (i_1, \ldots, i_k)$ denote a non-empty subsequence of indices $(1, 2, \ldots, d)$. Consider a set of latents $z \in \mathcal{Z}$, that instantiates a data-generating process (DGP) $g : \mathcal{Z} \rightarrow \mathcal{X}$ from the latents $z$ to observations $x$ and a labeling function $h : \mathcal{Z} \rightarrow \mathcal{Y}$ from latents $z$ to labels $y$. We assume observational sufficiency of the DGP: observations are sufficient for determining the label.

Definition 3.1. Mechanism. For a particular latent configuration $z \in \mathcal{Z}$, we say that $f(\cdot; \theta) : \mathcal{X} \rightarrow \mathcal{Y}$ uses mechanism $I$ on that example (where $I \subseteq [d_z]$ is the subset of indices of the latents) whenever $f(g(z'); \theta) = f(g(z); \theta)$ for all $z' \in \mathcal{Z} | z'_I = z_I$. Based on previous research, simplicity bias is where a model tends to rely on latent features which produce a ‘simpler’ decision boundary or solution. We call such latents corresponding to spurious correlations in the learned model spurious cues.
While a few recent analyses have addressed similar questions, they focus on in-distribution test datasets and coarse-grained measures like loss values [39, 31, 44]. In contrast, we emphasize out-of-distribution, counterfactual datasets. This is motivated from a notion of behavioral equivalence as the ideal goal of knowledge distillation. Our experiments are focused on the following questions for analysis:

1. For soft-target-matching-only distillation, only the simplest mechanisms will match.
2. Training a student model by minimizing the standard distillation objective is insufficient to guarantee knowledge transfer from teacher to student. Meanwhile, there exist distillation methods more likely to transfer all mechanisms.
3. Even if a teacher and student have similar fidelity (accuracy on the base task on the distillation set or even the teacher’s training set), they do not necessarily behave the same out of distribution [39].

The second point above is a consequence of the recent advances made in the field of Nonlinear Independent Component Analysis [68, 69, 70, 71, 72, 66] and disentangled representation learning [73, 74]. These demonstrate that producing the same outputs on a given dataset is insufficient to guarantee two models rely on the same underlying mechanisms for making their predictions. These suggest distillation is limited in what knowledge it can transfer—this depends on what data is shown to the models during distillation. We probe this further via empirical investigations.

4 Mechanistic evaluation of distillation

In this section we highlight the experimental setup and results for the two datasets as described in (Figure 1), for self-distillation on both ResNets and transformers.

4.1 Training and evaluation

Dataset generation. We follow prior work on understanding distillation, which primarily uses synthetic datasets to evaluate distillation protocols in a controlled manner [39, 41, 40, 45, 43, 42]. Having control over the data-generating process allows us to be precise about the distribution shift that occurs in the distillation dataset with respect to the teacher’s pretraining data, in order to evaluate a student model’s reliance on different mechanisms by altering the underlying latents. We assume a set of ‘natural’ latents underlie the labeling function $h$ of the data-generating process. All other latents are either uncorrelated with the label or model ‘spurious’ cues in the data. If using information from spurious latents leads to simpler functions, neural network simplicity bias [59, 57, 58, 55, 75, 76, 77, 78] suggests that a network will rely on them rather than the natural attributes for reducing the task loss. We denote the $n$ mechanisms defined by these spurious latents as $\{I_s_1, I_s_2, \ldots, I_s_n\}$. We design two datasets across both image and text data, called dominoes (images) and parity (language), shown in Fig. 1. These datasets have been used by prior works for modeling neural networks’ behavior regarding simplicity bias [57, 55], transfer learning [79, 80, 81, 82], disentangled representation learning [83], and scaling laws [84].

Training protocol. We use teacher and distillation datasets with different distributions over the latents, modeling the fact that a practitioner training a student model is unlikely to have access to the same dataset as the teacher model’s training data. To understand the effect of distribution shifts, we test our dominoes dataset (with an image mechanism and two spurious mechanisms) under all possible combinations of distillation and teacher dataset mechanisms (Figure 3).

Evaluation protocol. To evaluate whether a model uses any given mechanism, we randomise or remove latents corresponding to the content of the original image on the dominoes dataset, and report the expected divergence (as in Definition 3.1). For the image dominoes datasets, we remove the latents entirely.

4.2 Distribution shift and loss function effect on dominoes dataset

This section explores the effect of distribution shifts on a 3-mechanism image dataset for ResNet-18 self-distillation. We use each of 7 possible datasets where at least 1 of 3 mechanisms exists for teacher training, distillation and test evaluation. This gives 49 student/teacher mechanism combinations and 343 categorical final test values per loss function, each run with 3 seeds.
Figure 3: Final test accuracy. (a) For the ‘Similarity’ group, a lower bound is observed on performance across all combinations. (b) Jacobian loss has slightly lower performance when teacher and student do not share mechanisms (‘Neither’ group), improved performance for certain test mechanisms when only the teacher contains it (‘Teacher’, ‘Teacher SP’ groups) and reduced performance for certain test mechanisms when only the student contains it (‘Student’, ‘Student TP’ groups). (c) Contrastive loss strongly upper bounds test accuracy for spurious mechanisms in ‘Similarity’ group. Even when the student, teacher and test datasets share the spurious mechanism (‘Similarity’ group), learning is impeded. The only exception to this is the box mechanism in the test dataset, where simplicity bias is still observed.

Notation. \( S \) is distillation dataset (student) mechanism and \( T \) is teacher dataset mechanism. ‘Base distillation’ means softmax logit KL matching. All mechanisms are denoted by a single letter (see Figure 1) – \( I \): image (CIFAR-10), \( A \): spurious mechanism A (box), \( B \): spurious mechanism B (F-MNIST). In Figures 3 and 4, each strip corresponds to a different test mechanism, and each group to the relationship between \( S \) and \( T \). The meaning of the groups in the figure labels are as follows. Similarity: test mechanism overlaps completely with both student and teacher mechanisms. Student: test mechanism in \( S \) and not in \( T \). There must be a shared mechanism between \( S \) and \( T \), excluding the test mechanism. Teacher: test mechanism in \( T \) but not in \( S \), with same criteria for shared subtask as in Student group. Student TP: test mechanism is covered by \( S \) and shares a subset with the \( T \). Overlap: \( S, T \) share a subset and this subset is not in test mechanism. Neither: teacher shares no mechanisms with student. Any scenarios not fitting these categories are classified under Other, a broad class where the student and teacher each contain some subset of the test mechanism. This bound is highest when all three mechanisms are present (test mechanism IAB). This grouping does not use the relation between the test mechanism and \( S, T \). However, such relations are used in tables in the supplementary material. Finally, Figure 5 shows the mean and variance of final accuracy values for separated test mechanisms. Refer to supplementary material, Sections D and E for the rest of the results.

Base loss. Simplicity bias is observed in Figure 5 column 1 with base distillation. When the box mechanism (mechanism A) is present in the student and teacher datasets, it is learned while CIFAR-10 and F-MNIST are ignored. This is expected behaviour, as the teacher also shows simplicity bias. Interestingly, the student can learn a new F-MNIST mechanism (column 1, row 4 under test
mechanism B) if it is present in the distillation dataset and correlated with the box mechanism. This is an example of a type of ‘secondary transfer’ common in base distillation. If the teacher mechanism contains mechanisms P and Q, the student mechanism contains mechanisms Q and R, and we test on just mechanism P, the student may have high performance. This seems obvious, as the shared mechanism Q allows the teacher to produce correctly labelled examples. However, Jacobian and contrastive losses are less likely to produce this effect (Figure 5).

**Jacobian matching loss.** Values in Figure 5 with a change greater than 2 standard deviations are often in cases of distribution shift (i.e., the teacher and student datasets do not contain the same mechanism). In particular, there is decreased learning if the test mechanism was only in the distillation dataset. Jacobian loss is more likely to match \( S \) to \( T \) than base distillation if the test mechanism is in \( T \) but not in/partially in \( S \) (Table ??, column ‘in \( T \)’). Also, it is less likely to transfer all of \( S \) (Table ??, column ‘In \( S \)’) if it is not in/partially in the teacher dataset. KL divergence on test datasets decreases if the teacher and student datasets are identical (Figure ??). In contrast, KL divergence increases when \( S \neq T \) (e.g. mechanism I and AB) or one of \( T \) or \( S \) contains extra mechanisms that the other dataset did not. In this sense, Jacobian loss may not improve accuracy, but leads to better matching overall.

**Contrastive loss.** There is strong suppression of box mechanism transfer where it is not present in both teacher and student datasets (Figures 3 and 5). This transfer suppression is greatest when the spurious test mechanism is present in only one of the student or teacher datasets. Relative to performance in base distillation, this effect is surprisingly strong for the image corrupted by the box mechanism (Figure 5, test mechanism IA, ‘Student’/‘Teacher’ groups). This may model well behaviour on a type of spurious feature often present in realistic vision datasets. However, for all mechanisms, training with contrastive loss takes longer to achieve the same accuracy, resulting in a significant performance-robustness trade-off (supplementary material Table 1, column ‘\( EQ \)’). It
can also lead to transfer of simple mechanisms if they are present in both datasets. Contrastive loss produces the largest decrease in teacher-student output KL divergence compared to base and Jacobian distillation (Figure 4). The greatest KL divergence values for this loss function are for the ‘Neither’ group, where the teacher and student match but do not match the test mechanism. Contrastive loss seems to trade off matching on the support set of the teacher and student’s intersection for poorer performance entirely out of distribution of both.

Figure 5: Final accuracy/accuracy change and standard deviation on various test mechanisms, on dominoes dataset. Row labels within the heatmaps indicate student mechanism. The base distillation column gives raw values. The Jacobian and contrastive loss columns are differences, given by by new loss function minus base distillation. Middle column. The effect of Jacobian loss is subtle. It typically results in the greatest reduction in performance when the student dataset alone contains the test mechanism. Right column. Contrastive loss leads to reduction in transfer of the spurious mechanisms A and B (rows 2, 3) when both are present in the student and teacher datasets.

4.3 Fraction of spurious mechanism on parity dataset

Figure 6 shows the training steps required to achieve a given accuracy threshold as a function of the distillation dataset probability of simple task parity correlating with hard task parity. This is for a test dataset where only the hard task corresponds to the label. More hard task substrings to pick from increases the relative difficulty of learning the hard task mechanism, compared to the simple task. For full results, including for a test dataset where the simple and hard task both correspond to the
Figure 6: Steps to reach particular accuracy threshold on a test dataset vs distillation simple task fraction on the parity dataset. Test dataset mechanism: hard task always on, simple task randomised. Distillation dataset mechanism: hard task always on, and simple task probability on is given by $x$-axis. Teacher dataset mechanism: hard task only. When a given accuracy value is never obtained, an $x$ is marked and the datapoint is omitted from quadratic interpolation. Each data point is a separately trained student. Error bars show steps required in accuracy for $\pm 1$ standard deviation. (a) The model always learns the hard task, except when all distillation examples contain the spurious mechanism. Steps to reach a given accuracy for the hard task increases as fraction of simple mechanism in distillation dataset increases. This is expected for per-datapoint simplicity bias. (b) Compared to (a), for a given simple mechanism fraction, fewer training steps are required to reach a given accuracy.

5 Discussion

For all results in this section, the effect of changes such as adding loss terms will differ depending on the modality and dataset, hence the results here should not be considered general.

We observe simplicity bias in the base distillation accuracy in Figure 3 where all mechanisms containing spurious latents have higher maximum accuracy scores. Full results (supplementary material Sections D and E) show that the presence of the box mechanism in both teacher and student datasets will transfer the box mechanism to the student to near 100% accuracy, to the detriment of learning the image. In general, distillation does result in instances where simplicity bias can be avoided, especially if the student dataset does not contain the exact spurious mechanism that the teacher dataset does.

Jacobian matching loss on vision datasets has an effect of improved matching of the teacher mechanism, and reduced learning of newer mechanisms only present in the student. In general the results for this loss function are subtle, though the most statistically significant (2 standard deviations minimum) differences can be found for student and teacher datasets with little overlap (Figure 5). This could be
explained by the theory presented in related work (supplementary material Section A). We postulate that due to the complexity and subtlety of this effect, other methods such as matching input-activation Jacobian or using different datasets may produce more pronounced or qualitatively different results.

Across all teacher-student-test dataset triplets we tested, contrastive loss has the lowest teacher-student KL divergence, despite being the slowest to train. This effect also holds on the patterned box dataset, where not only the location but also the pixel values correspond to the label. However, there is an accuracy penalty of around 40% (supplementary material Section D, Table 1). This trade-off may be worthwhile in cases where it is important that the nature of the representation the student learns is similar to that of the teacher and large quantities of compute are available. Since prior work [31] shows that with a long enough distillation training, distillation effectiveness increases, there is no reason to believe that increasing epochs will not eliminate this accuracy penalty. We leave exploration of this phenomenon to further work.

6 Conclusion

In the datasets we examined, we found that Jacobian matching is useful when the teacher dataset is cleaner than the student dataset. Furthermore, we found that contrastive distillation results in a noticeable mitigation of simplicity bias. For both Jacobian and contrastive representation distillation, when the test mechanism either subsets, is a subset of, or only partially overlaps with the teacher and student mechanisms, transfer is reduced when compared to base distillation. In both cases, we also observe slower training. Distillation results are always stopped at a fixed number of epochs, so final accuracy may continue improving in these examples if the student is trained for longer.

Results on the parity dataset also agree with our simplicity bias hypothesis. On distillation datasets where the simple task always corresponds to the hard task’s label, the hard task will never be learned by the student. Jacobian matching has the strongest effect on this dataset. It speeds up transfer of the hard task from a clean teacher distilled on a student dataset with the simple task, as long as the dataset has some examples for which the hard task only is predictive of the label.

6.1 Further work

We suggest that further work investigates how exactly the model uses each of the mechanisms, potentially locating ‘circuits’ corresponding to localized computation of concepts in the network, as per recent interpretability literature [85, 86, 87, 88, 89]. In particular, the mechanism definition may be most useful when each latent dimension corresponds to ‘features’—for example, using the Fourier spectrum of image data [90] or gradient spectral clustering [84]. Such human imperceptible statistical correlations often form the backbone of how models learn algorithms to compute tasks [87, 90], leading to models vulnerable to adversarial attacks. A more modular representation should also allow our data-generating process to align better to how models process information.

Acknowledgements

We would like to express my gratitude to Herbie Bradley for valuable proofreading and contributions to the final presentation of this paper. Thanks also extend to Euan Ong and Rudolf Laine for their edits and insightful suggestions.

Author contributions are as follows: Cindy conducted and evaluated experiments and wrote the codebase for dominoes image experiments and modified Bruno’s parity experiment codebase. Theory and approach was motivated by literature review by Ekdeep, Robert, Cindy and Bruno, with input from David. Bruno wrote the codebase for the parity experiments. The concept of mechanisms and dominoes dataset design were provided by Ekdeep.

This paper represents the final outcome of a project that began with Cindy’s MEng dissertation and continued beyond. David played a key role as the Principal Investigator. The invaluable guidance of the co-authors, who also co-supervised the work, greatly contributed to its success. We would like to thank the Berkeley Existential Risk Initiative (BERI) for partially funding compute.
References


[90] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019.