# What Mechanisms Does Knowledge Distillation Distill?

**Cindy Wu**[*]
University of Cambridge

**Ekdeep Singh Lubana**
University of Michigan
CBS, Harvard University

**Bruno Kacper Mlodozeniec**
University of Cambridge

**Robert Kirk**
University College London

**David Krueger**
University of Cambridge

**Editors:** Marco Fumero, Emanuele Rodolà, Clementine Domine, Francesco Locatello, Gintare Karolina Dziugaite, Mathilde Caron

## 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, 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

[*]Correspondence to wu.cindyx@gmail.com

Proceedings of the I edition of the Workshop on Unifying Representations in Neural Models (UniReps 2023).

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, 45, 47], 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 $f : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^K$ 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 $f^{\texttt{SM}_\texttt{T}} = \mathrm{softmax}_T \circ f$, where $\mathrm{softmax}_T(z)_i = \frac{\exp z_i/T}{\sum_j \exp z_j/T}$ 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 $\mathcal{D}_\texttt{t}$ be the dataset used to train the teacher model $f_\texttt{t} : \mathbb{R}^n \times \mathbb{R}^{d_\texttt{t}} \to \mathbb{R}^k$ (whose parameters are $\theta_\texttt{t} \in \mathbb{R}^{d_\texttt{t}}$). The goal in knowledge distillation is to use this teacher model to train a 'student' model $f_\texttt{s} : \mathbb{R}^n \times \mathbb{R}^{d_\texttt{s}} \to \mathbb{R}^k$ by finding a set of parameters $\theta_\texttt{s} \in \mathbb{R}^{d_\texttt{s}}$ such that the outputs of the student model $f_\texttt{s}(\cdot;\theta_\texttt{s})$ match, in some specific sense, outputs from the teacher model $f_\texttt{t}(\cdot;\theta_\texttt{t})$ on a 'distillation dataset' $\mathcal{D}_\texttt{distill}$. 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 [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. [27], the standard distillation pipeline involves optimizing the agreement between teacher and student model predictions by minimizing the $\mathcal{KL}$-divergence between them:

$$\mathbb{E}_{x\sim\mathcal{D}_{\texttt{distill}}}\left[D_{\mathrm{KL}}\left(f_{\mathtt{t}}^{\mathtt{SM}_{\mathtt{T}}}(x;\theta_{\mathtt{t}})\|f_{\mathtt{s}}^{\mathtt{SM}_{\mathtt{T}}}(x;\theta_{\mathtt{s}})\right)\right] \tag{1}$$

The objective above is equivalent to minimising cross-entropy loss with $f_{\mathtt{t}}^{\mathtt{SM}_{\mathtt{T}}}(x;\theta_{\mathtt{t}})$ as the 'soft' targets for a student model. As shown by Hinton et al. [27], assuming the logits $f_{\mathtt{t}}(x), f_{\mathtt{s}}(x)$ are zero-centered (mean zero), in the high temperature limit $T\to\infty$ this is equivalent to minimising the average logit squared difference: $\|f_{\mathtt{t}}(x)-f_{\mathtt{s}}(x)\|^2$.

While a standard cross-entropy loss promoting correct classification is often added to the distillation objective, we follow recent works on understanding knowledge distillation [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: $f_{\mathtt{t}}^{\mathtt{SM}_{\mathtt{T}}}, f_{\mathtt{s}}^{\mathtt{SM}_{\mathtt{T}}}$ and the input-output Jacobians $\mathbf{J}_{f_{\mathtt{s}}^{\mathtt{SM}_{\mathtt{T}}}(\cdot,\theta_{\mathtt{s}})}, \mathbf{J}_{f_{\mathtt{t}}^{\mathtt{SM}_{\mathtt{T}}}(\cdot,\theta_{\mathtt{t}})}\in\mathbb{R}^{K\times n}$ match on examples in the distillation dataset $\mathcal{D}_{\texttt{distill}}$. 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 a the following penalty to the standard distillation loss of Eq. 1:

$$\|\mathbf{J}_{f_{\mathtt{s}}^{\mathtt{SM}_{\mathtt{T}}}(\cdot,\theta_{\mathtt{s}})}(\boldsymbol{x})-\mathbf{J}_{f_{\mathtt{t}}^{\mathtt{SM}_{\mathtt{T}}}(\cdot,\theta_{\mathtt{t}})}(\boldsymbol{x})\|_2^2 \tag{2}$$

Beyond distillation [61, 62], the objective above or variants of it have been introduced in several contexts, such as improving out-of-distribution generalization [63], improving adversarial robustness [64], and for learning disentangled representations [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 $\mathcal{D}_{\texttt{distill}}$. In [67], the authors propose to do this by maximising a lower-bound on the mutual information objective given below. Denote by $g_{\mathtt{s}}:\mathbb{R}^n\to\mathbb{R}^{h_{\mathtt{s}}}$, $g_{\mathtt{t}}:\mathbb{R}^n\to\mathbb{R}^{h_{\mathtt{t}}}$ the functions producing the penultimate layer features in the student and teacher models respectively.

$$\begin{array}{ll}\textbf{Teacher Representation:} & Z_{\mathtt{t}}=g_{\mathtt{t}}\left(X\right)\\ \textbf{Student Representation:} & Z_{\mathtt{s}}=g_{\mathtt{s}}\left(X\right)\end{array}\qquad\qquad X\sim\mathcal{D}_{\texttt{distill}}$$

$$\mathcal{MI}\left(Z_t,Z_s\right)\geq\mathbb{E}_{p(Z_{\mathtt{t}},Z_{\mathtt{s}})}[\log h(Z_{\mathtt{t}},Z_{\mathtt{s}})]+N\mathbb{E}_{p(Z_{\mathtt{t}})p(Z_{\mathtt{s}})}[\log(1-h(Z_{\mathtt{t}},Z_{\mathtt{s}}))] \tag{3}$$

where $h:\mathbb{R}^{h_{\mathtt{t}}}\times R^{h_{\mathtt{s}}}\to[0,1]$ 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 $p(Z_{\mathtt{t}},Z_{\mathtt{s}})$) or independently (from $p(Z_{\mathtt{t}})p(Z_{\mathtt{s}})$), assuming that they are sampled jointly $1/(N+1)$ of the time. $h$ typically takes the parametric form:

$$h(\boldsymbol{z}_{\mathtt{t}},\boldsymbol{z}_{\mathtt{s}})=\frac{\exp\boldsymbol{r}_{\mathtt{t}}^{\intercal}\boldsymbol{r}_{\mathtt{s}}/\tau}{\exp\boldsymbol{r}_{\mathtt{t}}^{\intercal}\boldsymbol{r}_{\mathtt{s}}/\tau+\frac{N}{|\mathcal{D}_{\texttt{distill}}|}},\qquad\begin{array}{l}\boldsymbol{r}_{\mathtt{t}}=W_{\mathtt{t}}\boldsymbol{z}_{\mathtt{t}}/\|W_{\mathtt{t}}\boldsymbol{z}_{\mathtt{t}}\|,\\ \boldsymbol{r}_{\mathtt{s}}=W_{\mathtt{s}}\boldsymbol{z}_{\mathtt{s}}/\|W_{\mathtt{s}}\boldsymbol{z}_{\mathtt{s}}\|,\end{array} \tag{4}$$

where $W_{\mathtt{s}}\in\mathbb{R}^{h_{\mathtt{inter}}\times h_{\mathtt{s}}}$, $W_{\mathtt{t}}\in\mathbb{R}^{h_{\mathtt{inter}}\times h_{\mathtt{t}}}$ are learnable parameters, and $\tau, h_{\mathtt{inter}}$ are hyperparameters.

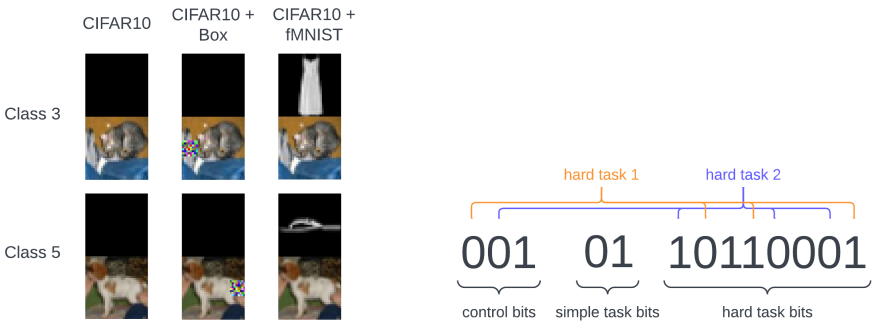



Figure 1: **Synthetic datasets.** Following the protocol above, we embed synthetic cues in two existing datasets: (1) Dominoes [57], where CIFAR-10 images are concatenated with F-MNIST images with the same class number. A third spurious mechanism selects a location of the CIFAR-10 image to set as randomised 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.

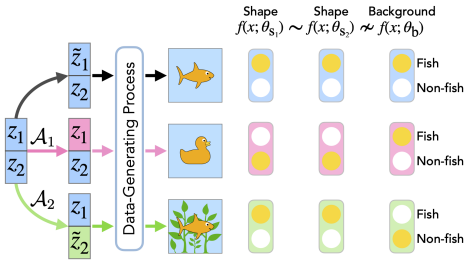



Figure 2: **Knowledge transfer.** We define successful knowledge transfer of a teacher and student model based on how they respond to unit interventions on the data-generating process, i.e., interventions on specific dimensions of the latent vector $z$; e.g., $\mathcal{A}_1$ (shape) and $\mathcal{A}_2$ (background) in the figure [35]. Here, yellow circles represent the prediction of a given model (column) on a counterfactual image (row). Models whose predictions are invariant to the same set of interventions (denoted $\theta_1 \sim \theta_2$) are termed mechanistically similar.

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

**Definition 3.1. Mechanism.** For a particular latent configuration $\boldsymbol{z} \in \mathcal{Z}$, we say that $f(.;\theta) : \mathcal{X} \to \mathcal{Y}$ uses mechanism $I$ on that example (where $I \subseteq [d_z]$ is the subset of indices of the latents) whenever $f(g(\boldsymbol{z}');\theta) = f(g(\boldsymbol{z});\theta)$ for all $\boldsymbol{z}' \in \mathcal{Z} | \boldsymbol{z}'_I = \boldsymbol{z}_I$.

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 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

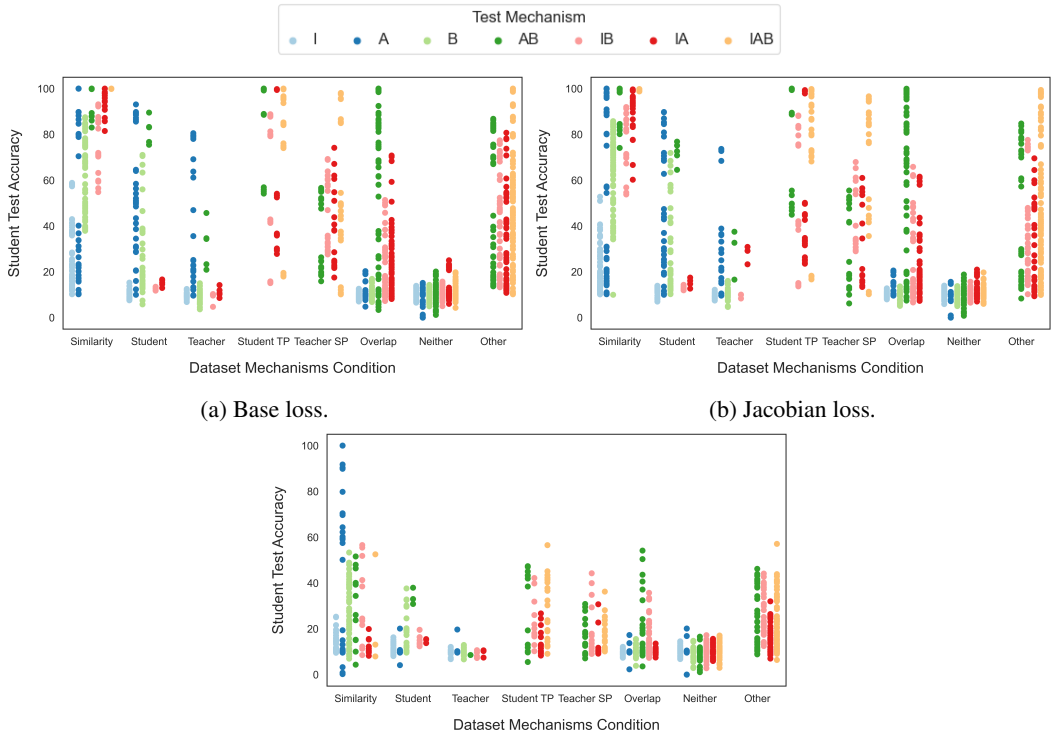
**Dataset generation.** We follow prior work on understanding distillation, which primarily uses synthetic datasets to evaluate distillation protocols in a controlled manner [39, 41, 40, 45, 43, 42]. Having control over the data-generating process allows us to be precise about the distribution shift 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, 35, 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, 35], 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.




(a) Base loss.

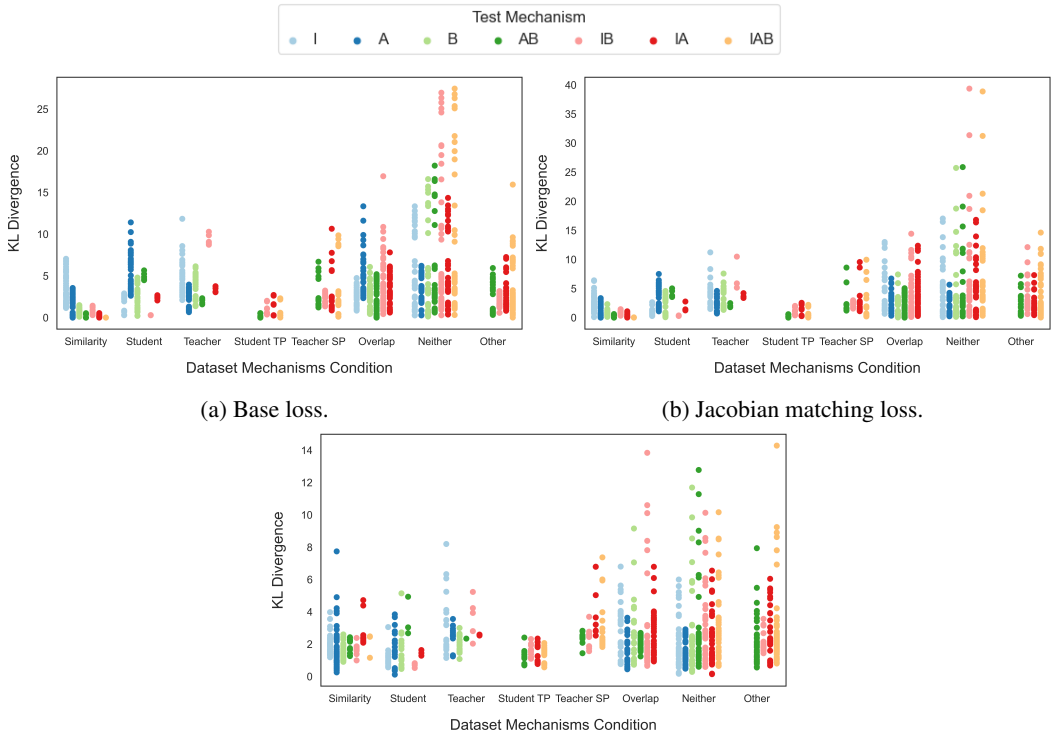
(b) Jacobian loss.

(c) Contrastive loss.

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




(a) Base loss.

(b) Jacobian matching loss.

(c) Contrastive loss.

Figure 4: **Final test KL divergence**. **(b)** Jacobian loss leads to especially high range of final KL divergence when the teacher and student do not share mechanisms. **(c)** Contrastive loss further bounds teacher-student KL divergence and results in most effective matching of teacher and student.

mechanism B) if it is present in the distillation dataset and correlated with the box mechanism. This is an example of a type 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 3, 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.

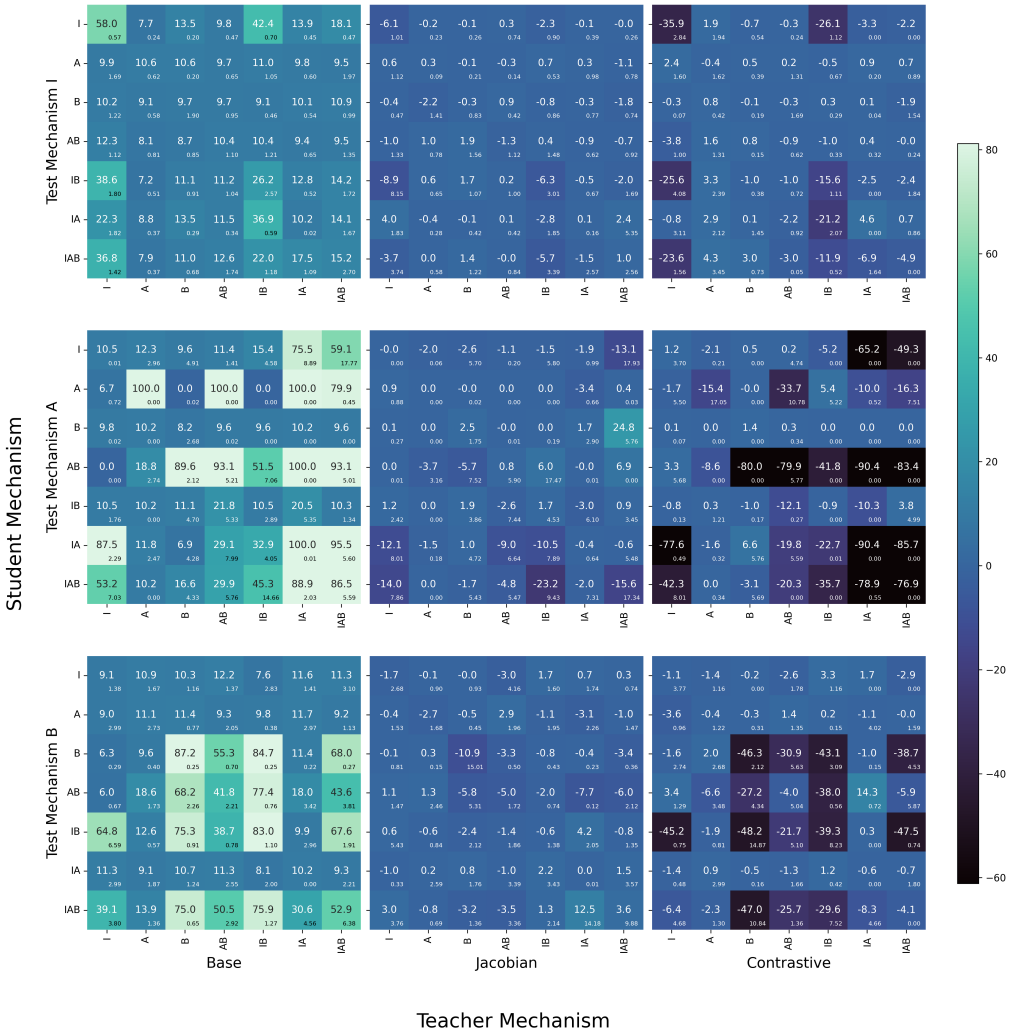
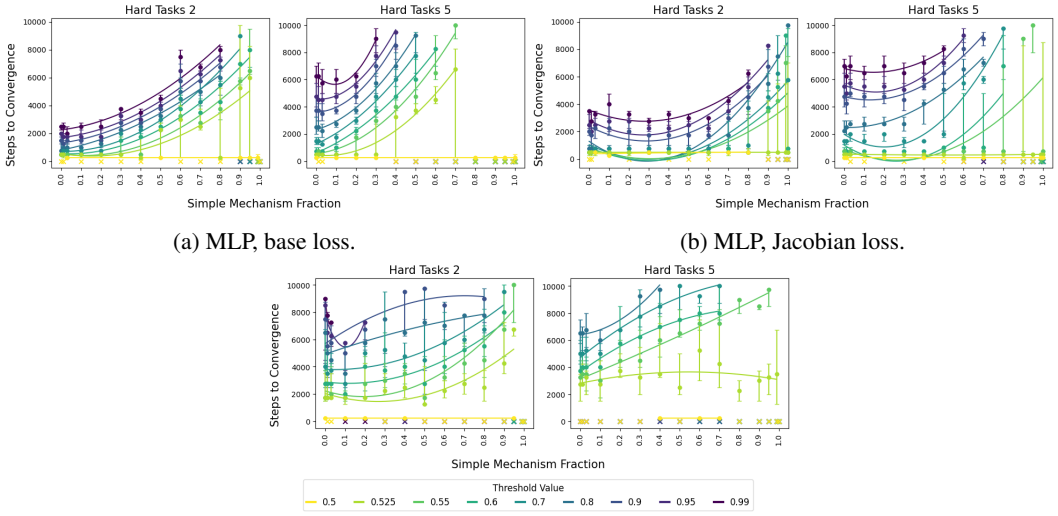



Figure 5: **Final accuracy/accuracy change and standard deviation on various test mechanisms**, on dominoes dataset. Row labels within the heatmaps indicate student mechanism. The base distillation column gives raw values. The Jacobian and contrastive loss columns are differences, given by by new loss 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




(a) MLP, base loss.

(b) MLP, Jacobian loss.

(c) Nano-GPT transformer classifier, base loss.

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, c)** 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.

label, and accuracy/entropy over training time on datasets with counterfactually randomized latents, see supplementary material Section F.

Except for the case where all distillation training samples have the simple task on, the model can always learn the hard task. This is true for both MLPs and transformers (Figure 6). The rate at which this task is learned is strongly affected by the fraction of samples with only the hard task present. The more hard tasks, the more training time is required to learn to a specific accuracy threshold for a fixed fraction of spurious mechanism. Adding a Jacobian loss term speeds up the learning of harder mechanisms present in both teacher and student datasets. This can be seen by comparing Figure 6(a) to (b): the student reaches higher evaluation accuracy within a fixed number of steps on the dataset with only hard task corresponding to the label.

## 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


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

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

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[4] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

[6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

[8] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[10] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*, 2023.

[11] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[12] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[13] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[15] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[16] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[18] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[19] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[20] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[21] Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*, 2022.

[22] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[23] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

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

[26] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[28] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[29] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[30] Zhengxuan Wu, Atticus Geiger, Josh Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah D Goodman. Causal distillation for language models. *arXiv preprint arXiv:2112.02505*, 2021.

[31] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10925–10934, 2022.

[32] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.

[33] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

[34] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.

[35] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. *arXiv preprint arXiv:2211.08422*, 2022.

[36] Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*, 2021.

[37] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

[38] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[39] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021.

[40] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR, 2019.

[41] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12925–12935, 2020.

[42] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[43] Huan Wang, Suhas Lohit, Michael N Jones, and Yun Fu. What makes a" good" data augmentation in knowledge distillation-a statistical perspective. *Advances in Neural Information Processing Systems*, 35:13456–13469, 2022.

[44] Samira Abnar, Mostafa Dehghani, and Willem Zuidema. Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555*, 2020.

[45] Gal Kaplun, Eran Malach, Preetum Nakkiran, and Shai Shalev-Shwartz. Knowledge distillation: Bad models can be good role models. *arXiv preprint arXiv:2203.14649*, 2022.

[46] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[47] Vaishnavh Nagarajan, Aditya Krishna Menon, Srinadh Bhojanapalli, Hossein Mobahi, and Sanjiv Kumar. On student-teacher deviations in distillation: does it pay to disobey? *arXiv preprint arXiv:2301.12923*, 2023.

[48] Umang Gupta, Jwala Dhamala, Varun Kumar, Apurv Verma, Yada Pruksachatkun, Satyapriya Krishna, Rahul Gupta, Kai-Wei Chang, Greg Ver Steeg, and Aram Galstyan. Mitigating gender bias in distilled language models via counterfactual role reversal. *arXiv preprint arXiv:2203.12574*, 2022.

[49] Hugging Face Co. *DistilGPT2*, 2020. https://huggingface.co/distilgpt2.

[50] Hugging Face Co. *Distil* repository.*, 2020. https://github.com/huggingface/transformers/tree/main/examples/research_projects/distillation.

[51] Vinay Sisodia. *Distillation of CLIP model and other experiments*, 2021. https://tech.pic-collage.com/distillation-of-clip-model-and-other-experiments-f8394b7321ce.

[52] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proc. Euro. Conf. on Computer Vision (ECCV)*, 2018.

[53] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

[54] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint. arXiv:2006.09994*, 2020.

[55] Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Adv. in Neural Information Processing Systems (NeurIPS)*, 2020.

[56] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research (JMLR)*, 2020.

[57] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Adv. in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint. arXiv:1805.08522*, 2018.

[59] Preetum Nakkiran, Dimitris Kalimeris, Gal Kaplun, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Adv. in Neural Information Processing Systems (NeurIPS)*, 2019.

[60] Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will dnns choose? a study from the parameter-space perspective. *arXiv preprint. arXiv:2110.03095*, 2021.

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

[62] Suraj Srinivas and Francois Fleuret. Local affine approximations for improving knowledge transfer. *Idiap-Com Idiap-Com-01-2018, Idiap*, 3, 2018.

[63] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton Van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16761–16772, 2022.

[64] Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

[65] Edmond Cunningham, Adam Cobb, and Susmit Jha. Principal manifold flows. *arXiv preprint arXiv:2202.07037*, 2022.

[66] Luigi Gresele, Julius Von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? *Adv. in Neural Information Processing Systems (NeurIPS)*, 2021.

[67] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Representation Distillation, January 2022.

[68] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Adv. in Neural Information Processing Systems (NeurIPS)*, 2016.

[69] Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[70] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[71] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *Int. conf. on artificial intelligence and statistics (AISTATS)*, 2021.

[72] Luigi Gresele, Paul K Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. In *Uncertainty in Artificial Intelligence (UAI)*, 2020.

[73] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. int. conf. on machine learning (ICML)*, 2019.

[74] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020.

[75] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Adv. in Neural Information Processing Systems (NeurIPS)*, 2021.

[76] Haruki Watanabe. Counting rules of Nambu–Goldstone modes. *Annual Review of Condensed Matter Physics*, 2020.

[77] Sumio Watanabe. Almost all learning machines are singular. *2007 IEEE Symposium on Foundations of Computational Intelligence*, pages 383–388, 2007.

[78] Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A Louis. Is sgd a bayesian sampler? well, almost. *Journal of Machine Learning Research*, 22(79):1–64, 2021.

[79] Polina Kirichenko, Pavel Izmailov, Nate Gruver, and Andrew Gordon Wilson. On feature learning in the presence of spurious correlations. *arXiv preprint. arXiv:2210.11369*, 2022.

[80] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint. arXiv:2204.02937*, 2022.

[81] Puja Trivedi, Danai Koutra, and Jayaraman J Thiagarajan. A closer look at model adaptation using feature distortion and simplicity bias. *arXiv preprint arXiv:2303.13500*, 2023.

[82] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint. arXiv:2110.11328*, 2021.

[83] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *In Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

[84] Eric J Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *arXiv preprint arXiv:2303.13506*, 2023.

[85] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[86] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023.

[87] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023.

[88] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks, 2023.

[89] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability, 2023.

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