Learning from Teaching Regularization: Generalizable Correlations Should be Easy to Imitate

Can Jin[1*] **Tong Che**[2*] **Hongwu Peng**[3†]

Yiyuan Li[4†] **Dimitris N. Metaxas**[1‡] **Marco Pavone**[5‡]

[1]Rutgers University [2]Nvidia Research [3]University of Connecticut
[4]University of North Carolina at Chapel Hill [5]Stanford University
can.jin@rutgers.edu, tongc@nvidia.com

Abstract

Generalization remains a central challenge in machine learning. In this work, we propose *Learning from Teaching* (**LoT**), a novel regularization technique for deep neural networks to enhance generalization. Inspired by the human ability to capture concise and abstract patterns, we hypothesize that generalizable correlations are expected to be easier to imitate. LoT operationalizes this concept to improve the generalization of the main model with auxiliary student learners. The student learners are trained by the main model and, in turn, provide feedback to help the main model capture more generalizable and imitable correlations. Our experimental results across several domains, including Computer Vision, Natural Language Processing, and methodologies like Reinforcement Learning, demonstrate that the introduction of LoT brings significant benefits compared to training models on the original dataset. The results suggest the effectiveness and efficiency of LoT in identifying generalizable information at the right scales while discarding spurious data correlations, thus making LoT a valuable addition to current machine learning. Code is available at https://github.com/jincan333/LoT.

1 Introduction

Improving the generalization performance of models on unseen data is a major challenge in machine learning [6, 7, 73, 79, 107]. Despite its significant advances, identifying the most generalizable model within the vast space of potential models remains challenging. Existing deep learning approaches focus on crafting the hypothesis spaces where prediction errors are optimized using training data [33, 69, 71]. These spaces are shaped by inductive biases [33, 70] embedded in the neural architectures which include implicit assumptions about the data [1, 25, 95], objective functions (notably regularizers) [20, 68, 103], and learning methodologies [14, 72, 87].

In this paper, to enhance generalization, we use the methodology of regularization [37, 51, 88], which prioritizes specific regions in the hypothesis spaces. Regularization techniques often involve employing auxiliary losses or regularizers [20, 38, 103] alongside the primary task losses. For instance, L1 regularization [41, 92, 93] encourages sparsity within models [16, 40, 54, 57]. Other regularization techniques include model averaging [44, 102], dropout techniques [37, 67, 100], and additional

[0*]Equal contribution, [†]Equal contribution, [‡]Equal advising, Correspondence to: Can Jin <can.jin@rutgers.edu>, Tong Che <tongc@nvidia.com>.

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

optimization components [43, 63, 104]. Due to its effectiveness and simplicity, regularization is critical in modern machine learning techniques for achieving better generalization [37, 109].

We aim to answer the research question: *Among all possible models fitting the training data, which ones are inherently generalizable?* A common belief in cognitive science is that human intelligence development involves distilling information and filtering out extraneous details to discern 'simple' correlations among a few selected relevant abstract variables [18, 94]. This approach leads to the formation of correlations through simple patterns [2, 56] at the right scales. However, identifying simple correlations in deep learning remains challenging, mostly due to not being easy to identify the right scale of the problem. Studies in emergent languages suggest that the more structured a language is, the more efficiently it can be transmitted to message receivers [11, 56]. Inspired by this finding, we propose defining simple and generalizable correlations at the right scales, as those that can be readily imitated by other learners, provided they possess suitable inductive biases.

Based on this definition, we propose a novel regularization approach, *Learning from Teaching* (**LoT**). The core of LoT is to compute a measure of 'imitability' for the main model to learn data correlations at the correct scales. By adding this measure to the objective function and optimizing it during training, we encourage the teacher model to refine its learned multiscale correlations, making them more accessible through teaching, which in turn leads to better generalization. LoT computes this measure by jointly training the main model as the 'teacher' with one or more auxiliary 'student' models. The student models strive to distill and assimilate the correlations acquired by the teacher model. Thus, the learning performance of the student defines the measure of imitability of the teacher, which is then used as the LoT regularizer.

We conduct comprehensive experiments using LoT to improve the Reinforcement Learning (RL) formulation, as well as in Natural Language Processing (NLP) and Computer Vision (CV) applications. In RL, the experimental results demonstrate that LoT attains an average normalized reward enhancement of 44% on four Atari games. In language modeling tasks, LoT achieves significant perplexity reductions on the Penn Tree Bank [64] and WikiText-103 [65]. Notably, LoT enhances the supervised fine-tuning performance of LLaMA [96, 97] models on GSM8K [19] and MATH [35]. In image classification tasks, LoT achieves accuracy gains of 1.99% and 0.83% on CIFAR-100 [49] and ImageNet-1K [23], respectively.

2 Methodology

2.1 Generalizable and Spurious Correlations

Given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ generated from a data-generating distribution $\hat{D}$, there are infinitely many continuous functions f such that $f(\mathbf{x}) = y$ for all $(\mathbf{x}, y) \in \mathcal{D}$. Therefore, finding the f that precisely models the true generalizable correlation between $\mathbf{x}$ and y is challenging, especially with real-world data like natural images, which are complex and multiscale. In such scenarios, a neural network may compute incorrect (according to the ground-truth relationship between variables) yet perfect (in the empirical data distribution) correlations that explain the relationship between $\mathbf{x}$ and y [32, 73]. This phenomenon is particularly evident when y is entirely noise-based and independent of $\mathbf{x}$, but the neural network still fits y to $\mathbf{x}$ perfectly [73, 107]. This process, often called brute-force memorization [2, 13], involves the network creating intricate computational strategies to encode all $(\mathbf{x}, y)$ pairs in the samples. Consequently, correlations established in this way are spurious, originating from sampling noise in the data rather than ground-truth relationships.

But how do humans distinguish generalizable correlations from spurious ones? Instead of relying on brute-force memorization to establish input-output correspondences, humans naturally focus on understanding high-level concepts within the input data, selectively ignoring irrelevant details [18, 94]. This approach leads to the formation of correlations through simple, comprehensible patterns [2, 11]. Empirical evidence in emergent languages also suggests that the more compositional a language is, the more learners will use it [11, 56].

We can, therefore, define the distinctions between generalizable and spurious correlations. First, generalizable correlations are simple and comprehensible, exhibiting lower Kolmogorov Complexity [31, 58, 90]. Second, while there is only one ground-truth correlation for a dataset, the number of spurious correlations can be massive. These two major distinctions lead to the following hypothesis.

Hypothesis: Generalizable correlations should be more easily imitable by learners compared to spurious correlations. Specifically, assume T_G and T_S are two teacher models that capture the generalizable correlation and spurious correlation from a dataset, respectively. We have student learners S_G and S_S that separately imitate T_G and T_S:

- From an effectiveness perspective, the final training and test losses of learner S_G after training are typically lower than those of learner S_S.
- From an efficiency perspective, during training, the test losses of learner S_G decrease more rapidly than those of S_S.

This hypothesis emphasizes that generalizable correlations inherent in data are not only more interpretable but also more readily imitable. It suggests that the inherent simplicity and uniqueness of generalizable correlations make them more attainable and recognizable for learning algorithms, in contrast to the complex and abundant nature of spurious correlations derived from noise. In the following we present our novel approach.

2.2 Learning from Teaching Regularization

Building upon the Hypothesis, we propose that the ease of imitation of the teacher model by student models can serve as a proxy for the generalizability of learned representations. By measuring the 'imitability' of the teacher model in the learning process, we can infer the generalizability of it. A teacher that is easier to imitate implies higher generalization. We then design a novel regularization approach that involves training a teacher model T alongside student models S to imitate T, subsequently measuring the imitability of the teacher during training. We maximize imitability by incorporating it as an additional loss during the training of the teacher T. This imitability loss is termed the Learning from Teaching regularizer (LOT regularizer). By doing so, T is optimized to be a teacher that is easier to imitate and, thus, possesses superior generalization compared to models without the LOT regularizer. We refer to this class of regularization methods as 'Learning from Teaching Regularization' (LOT). LOT aligns with the broader concept of regularization in machine learning, where the goal is to promote generalizable representations and prevent overfitting.

Although LOT can be applied to supervised, unsupervised, and reinforcement learning, we begin our discussion with supervised learning. We train a network $T_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta}$, as the main model, which also serves as the teacher model. Additionally, we train a set of K networks $S_i, i = 1, 2, \cdots, K$, as the student models[1]. The total set of parameters of the K networks is denoted by $\boldsymbol{\phi}$. Given a training dataset $\mathcal{D}_t = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$, we train T and S to model $p(y|\mathbf{x})$, denoted as $p_t(y|\mathbf{x})$ and $p_s(y|\mathbf{x})$, respectively. Additionally, LOT includes a predefined imitability metric $\mu_{s,t}(\cdot) = \mu(S(\cdot), T(\cdot))$. Intuitively, $\mu_{s,t}$ measures the difference between S and T's predictions on the same input (occasionally denoted as μ henceforth for convenience). There are many possible choices for the metric μ, such as the L^2 loss between the hidden representations of a specific layer. In our experiments, we choose $\mu(\mathbf{x}) = \mu_{\text{KL}}(p_s(y|\mathbf{x})||p_t(y|\mathbf{x}))$, which is the KL-divergence [21], to quantify the distribution similarity between S and T.

We first train the teacher model. The objective function of the teacher combines the regular task loss with the additional LOT regularizer $R(\boldsymbol{\theta})$ (defined in Equation 3). For example, in supervised learning, we can use the negative log-likelihood loss for the regular task loss, and the objective function can be written as:

$$L_t(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{D}_t|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_t} \log p_t(y_i|\mathbf{x}_i) + R(\boldsymbol{\theta}), \tag{1}$$

where $|\mathcal{D}_t|$ is the number of samples in the dataset $\mathcal{D}_t$.

To train the student networks and enhance information diversity, we require an independent unlabelled dataset, denoted as $\mathcal{D}_s = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$. This dataset can be identical to $\mathcal{D}_t$, or generated either by a generative model trained on $\mathcal{D}_t$ or through alternative augmentation methods (e.g. synthetic data generation). This unlabelled dataset constitutes the environment for the student networks to follow the prediction of the teacher and, therefore, explores and generalizes beyond the original training data.

[1]For convenience, S is referred to as a single student learner henceforth.

Specifically, the student networks' goal is to imitate the correlations acquired by the teacher network during the training process. The training loss for students can be written as:

$$L_s(\phi) = \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \sum_{i=1}^{K} \mu_{s_i,t}(\mathbf{x}), \tag{2}$$

where $|\mathcal{D}_s|$ is the number of samples in the unlabelled dataset $\mathcal{D}_s$. The loss function L_s encourages the student networks to learn from the teacher network by minimizing the difference between their predictions, as measured by the metric $\mu_{s,t}(\mathbf{x})$.

The feedback from all students S_i constitutes the LOT regularizer:

$$R(\boldsymbol{\theta}) = \frac{\alpha}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \sum_{i=1}^{K} \lambda_i \mu_{t,s_i}(\mathbf{x}), \tag{3}$$

where $\lambda_i \geq 0$ represents the coefficient weight of the i-th student S_i, with $\sum_{i=1}^{K} \lambda_i = 1$. The λ_i can be either a learnable parameter or fixed, such as $\frac{1}{K}$. Essentially, the LOT regularizer measures the imitability of the teacher. The regularization coefficient α controls the trade-off between the original task learning objective of T and the feedback from the students.

The detailed procedure of LOT for supervised and unsupervised learning is outlined in Algorithm 1, and LOT regularization for RL (using PPO as an example) is outlined in Algorithm 2. The teacher T and student S_i networks are initialized differently to ensure they learn diverse features and representations. In both algorithms, the teacher and student networks iteratively learn from each other, with the students imitating the teacher's correlations and the teacher incorporating the students' feedback into the learning process.

Algorithm 1 Learning from Teaching Regularization

Input: Dataset $\mathcal{D}_s, \mathcal{D}_t$, Regularization Coefficient $\alpha > 0$, Student Steps Ratio $N > 0$
Initialize teacher network T parameterized by $\boldsymbol{\theta}$ and student networks $S_i, i = 1, 2, \cdots K$, parameterized by $\boldsymbol{\phi}$.
repeat
  Sample a batch of data $\mathcal{B}_t \subset \mathcal{D}_t, \mathcal{B}_s \subset \mathcal{D}_s$
  Compute $\tilde{R}(\boldsymbol{\theta}) = \frac{\alpha}{|\mathcal{B}_s|} \sum_{\mathbf{x} \in \mathcal{B}_s} \sum_{i=1}^{K} \lambda_i \mu_{t,s_i}(\mathbf{x})$
  Compute $\tilde{L}_t(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{B}_t|} \sum_{(\mathbf{x},y) \in \mathcal{B}_t} \log p_t(y|\mathbf{x}) + \tilde{R}(\boldsymbol{\theta})$
  Update $\boldsymbol{\theta}$ using gradient $\nabla_{\boldsymbol{\theta}} \tilde{L}_t(\boldsymbol{\theta})$
  for $i = 1$ **to** N **do**
    Sample $\mathcal{B}_s \subset \mathcal{D}_s$
    Compute $\tilde{L}_s(\boldsymbol{\phi}) = \frac{1}{|\mathcal{B}_s|} \sum_{\mathbf{x} \in \mathcal{B}_s} \sum_{i=1}^{K} \mu_{s_i,t}(\mathbf{x})$
    Update student networks' parameters $\boldsymbol{\phi}$ using loss gradient $\nabla_{\boldsymbol{\phi}} \tilde{L}_s(\boldsymbol{\phi})$
  end for
until T converges

2.3 Discussion

The works most related to LOT are knowledge distillation (KD) [29, 36] and ease-of-teaching [11, 56] in emergent languages. However, LOT differs significantly from these approaches. In KD, a teacher model containing task-specific knowledge transmits this knowledge to a student model (often smaller than the teacher), with the primary focus on the student's performance post-distillation. Conversely, in LOT , both the teacher and student models may lack or possess different task-specific knowledge. Generalization is improved through joint training, incorporating additional signals from student feedback. In emergent languages, Li and Bowling [56] propose that structured language is easier to teach to other agents than less structured ones, achieving higher task success rates with less training. Additionally, Chaabouni et al. [11] identify a strong positive correlation between language transmission efficiency to new message receivers and the degree of compositionality (structuredness) of the language. In LOT , we focus on tasks distinct from emergent languages, finding that generalizable correlations are easier to imitate. Under our Hypothesis, we design a novel

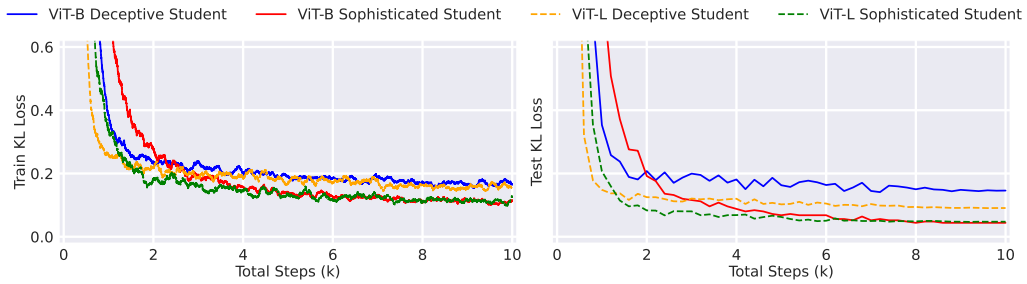

Figure 1: Training and test KL-divergence losses of student models in LOT using ViT-B/16 and ViT-L/16 on CIFAR-100 with different teacher models. The sophisticated students achieve lower losses than the deceptive students given the same computational budget.

LOT regularizer and algorithm to enhance the generalization of deep neural networks, extending the ease-of-teaching concept to supervised, unsupervised, and reinforcement learning. In parallel work, Ning et al. [74] proposes Learning by Teaching (LbT), which utilizes teacher and student models to generate answers as training samples for the teacher model. However, the regularization method in LOT is fundamentally distinct from that in Ning et al. [74].

3 Experiments

We first validate our Hypothesis in Section 3.1. Subsequently, we assess the performance of LOT across several tasks: Atari games (Section 3.2), language modeling (Section 3.3), and image classification (Section 3.4). We compare LOT to a Teacher-only baseline, wherein the regularization coefficient α in $R(\boldsymbol{\theta})$ is set to 0, thereby blocking the student feedback. Unless specified otherwise, we employ only one student model. Except for the Atari games where the student can learn from the offline samples of the teacher, we set $N = 1$ to manage computation (we study the impact of N in Section 3.6). Moreover, we study the computational efficiency and effects of hyperparameters of LOT in Sections 3.5 and 3.6.

3.1 Generalizable Correlations are Easier to Imitate than Spurious Correlations.

In our Hypothesis, learners are presumed to more readily imitate generalizable correlations than spurious ones. To investigate this, we design experiments involving two distinct teacher models: a sophisticated teacher and a deceptive teacher. The sophisticated teacher effectively captures generalizable correlations, while the deceptive teacher primarily learns spurious correlations. We use an identical student model to learn from both teachers separately, monitoring the student-teacher KL divergence during training and testing. The student that learns easier-to-imitate correlations is expected to exhibit lower training and test KL losses with fewer training steps.

We employ the ViT-B/16 and ViT-L/16 architectures [24] for both the teachers and students. The sophisticated teachers are trained on the full CIFAR-100 [49] training set for $10{,}000$ steps to achieve optimal convergence. The deceptive teachers, using the same hyperparameters and training steps as the sophisticated teachers, are trained on a random subset of $2,560$ images from the CIFAR-100 training set, leading to over-fitting. Consequently, the sophisticated teachers are expected to exhibit better generalization ability (their test accuracy surpasses that of the deceptive teachers by 14%).

The two student models referred to as the sophisticated student and the deceptive student, share identical hyperparameters and initializations. They are trained to imitate the correlations from their respective teachers on the full CIFAR-100 training set. The teacher models are kept frozen during the training of the students, with the objective $L_s(\boldsymbol{\phi})$ defined as follows:

$$L_s(\boldsymbol{\phi}) = \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x} \in \mathcal{D}_s} \mu_{\text{KL}}(p_s(y|\mathbf{x}) || p_t(y|\mathbf{x})), \tag{4}$$

where $\mathcal{D}_s$ represents the full training set of CIFAR-100.

We present the training and test losses in Figure 1 and make the following observations:

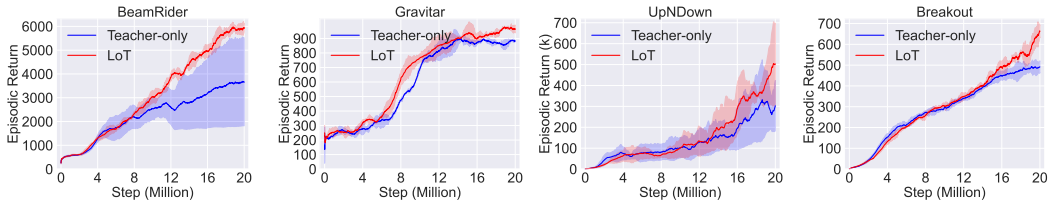

Figure 2: The episodic return of the teacher agent in LOT and the Teacher-only on four Atari games (averaged over ten runs). LOT demonstrates return gains over Teacher-only on all games.

- Given the same computational budget, the sophisticated students achieve lower final KL losses on both the training and test sets compared to the deceptive students. This suggests that the student can more effectively imitate the prediction distribution of a teacher that captures generalizable correlations.
- The deceptive students require more training steps to achieve the same training and test student-teacher KL losses as the sophisticated students. This indicates that learners tend to grasp spurious correlations much more slowly than generalizable correlations.

These results suggest that generalizable correlations are easier to imitate than spurious ones. In LOT , we expect the teacher model to master generalizable correlations by incorporating feedback from students via the LOT regularizer.

3.2 Atari Games

We conduct experiments on four Atari games, namely BeamRider, Breakout, UpNDown, and Gravitar, following the implementation in Huang et al. [42]. Both the LOT and Teacher-only agents have identical hyperparameters. All agents are trained using Proximal Policy Optimization (PPO) [83]. While the teacher agents interact with the game environment, the student agents are trained on the **most recent** $10{,}240$ **samples** generated by the teacher agents, ensuring that LOT and Teacher-only experience the same environmental interactions. We use different α values for various games and set $N = 5$ to efficiently imitate the teacher. More details are provided in Appendix D.

The empirical results are presented in Figure 2, and we make the following observations:

- LOT improves the agent return compared to the Teacher-only version with 20 million teacher training steps. Specifically, LOT achieves $\{63.14\%, 9.79\%, 66.48\%, 35.70\%\}$ normalized return enhancements on {BeamRider, Gravitar, UpNDown, Breakout}.
- The performance gain of LOT becomes more prominent as the training progresses (from 15 million to 20 million steps).

These results suggest that LOT is an effective approach for enhancing the generalization of RL agents, as it requires no additional environmental interactions while delivering significant performance gains.

3.3 Language Modeling

Language modeling is a widely acknowledged NLP task, and regularization techniques have been demonstrated to significantly enhance performance in this domain [106]. To examine the impact of LOT on language modeling, we conduct experiments in two scenarios: unsupervised language pretraining and supervised fine-tuning.

3.3.1 Unsupervised Language Pretraining

We conduct experiments of LOT and Teacher-only using LSTM [39], AWD-LSTM [66], and Transformer-XL [22] for teacher and student on Penn Tree Bank (PTB) [64] and WikiText-103 [65]. We follow the implementations outlined in Dai et al. [22], Merity et al. [66], Zaremba et al. [106]. In LOT , we utilize different coefficients α for various architectures and benchmarks to control the LOT regularizer. To ensure a fair comparison, we maintain the same total number of training steps

Table 1: The test perplexity of the teacher model in LOT and the baseline on PTB and WikiText-103. Results are averaged over three runs. LOT achieves consistent perplexity reduction over different choices of architectures and benchmarks.

Dataset	Teacher	Student	Teacher #Param.	Teacher-only	LOT
PTB	LSTM	LSTM	20M	82.75 ± 0.36	**71.72** ± 0.54
	AWD-LSTM	AWD-LSTM	24M	58.69 ± 0.37	**53.31** ± 0.56
WikiText-103	Transformer-XL-B	Transformer-XL-B	151M	23.72 ± 0.41	**21.65** ± 0.38
	Transformer-XL-L	Transformer-XL-L	257M	18.50 ± 0.25	**16.47** ± 0.23

(with teacher and student training steps accumulated) for LOT and the Teacher-only setup. Please refer to Appendix D for more implementation details.

From the empirical results presented in Table 1, we observe that LOT achieves notable perplexity (PPL) gains across various architectures and benchmarks under the same number of learning steps as Teacher-only. Specifically, LOT achieves at least 2 points PPL gains across all settings, and a 11.03 gain for LSTM on PTB. It indicates that LOT can be effectively applied to both LSTM and Transformer architectures in language pretraining.

3.3.2 Supervised Fine-tuning

Furthermore, to evaluate the effectiveness of LOT in fine-tuning pretrained large language models (LLMs), we conduct supervised fine-tuning (SFT) experiments using LLaMA-1 [96] and LLaMA-2 [97] on two mathematical reasoning benchmarks: GSM8K [19] and MATH [35].

We compare LOT to in-context learning (ICL) [9] and SFT. Following Touvron et al. [97], the number of in-context examples is 8 for GSM8K and 4 for MATH. The SFT configuration follows Yue et al. [105], and we fine-tune the LLaMA models for four epochs. In LOT , the teacher and student models share the same architecture for simplicity. The models are trained for two epochs in LOT to match the total training steps in SFT for fair comparison. All other configurations are consistent with those used in SFT. More implementation details are described in Appendix D.

We measure the accuracy of greedy decoding results in Table 2, and we observe that LOT enhances reasoning abilities on all architecture and dataset choices. This indicates the competence of LOT in improving the fine-tuning performance with a computational cost comparable to SFT.

Table 2: The accuracy of the teacher model in LOT and the baseline on GSM8K and MATH. Results are averaged over three runs.

Setting	GSM8K	MATH
LLaMA-1 $7B_{+ICL}$	10.69 ± 0.87	2.84 ± 0.25
LLaMA-1 $7B_{+SFT}$	34.39 ± 1.28	4.78 ± 0.23
LLaMA-1 $7B_{+LOT}$	**36.42** ± 1.46	**5.39** ± 0.28
LLaMA-2 $7B_{+ICL}$	14.62 ± 0.96	2.46 ± 0.25
LLaMA-2 $7B_{+SFT}$	39.81 ± 1.34	5.79 ± 0.31
LLaMA-2 $7B_{+LOT}$	**41.87** ± 1.62	**6.28** ± 0.22

3.4 Image Classification

To investigate the effects of LOT on computer vision tasks, we apply LOT to image classification by conducting experiments using ResNets [34], MobileNetV2 [81], ViT [24], and Swin [61] architectures pretrained on ImageNet-1K and ImageNet-21K [23] as teacher and student models. We choose CIFAR-100 [49] and ImageNet-1K as the downstream datasets. The total training steps for LOT and the Teacher-only approach are the same for a fair comparison. Further implementation details are provided in Appendix D. We conclude the following observations from results in Table 3:

- LOT achieves accuracy gains across various architectures and datasets without additional computational costs. For example, LOT improves test accuracy by almost 2 points using a ResNet-18 teacher and a ResNet-50 student on CIFAR-100 after pretrained on ImageNet-1K. Similarly, on the larger-scaled ImageNet dataset ImageNet-21K, LOT still obtains nearly 1 point improvement using ViT-B/16 as the teacher and ViT-L/16 as the student.
- The generalization of teacher models can be effectively enhanced by students of larger sizes. For instance, ResNet-50, ViT-L/16, and Swin-L students can enhance the performance of ResNet-18, ViT-B/16, and Swin-B teachers, respectively. Similarly, small student models

can also enhance the generalization performance of larger teacher models using LOT . For example, a MobileNetV2 student improves the performance of RestNet-18 and ResNet-50 by more than 1 point on CIFAR-100 with a much smaller model size. Similar results appear on the ViT-L/16 teacher and ViT-B/16 student combination in the ImageNet-1K task.

- For transformer-based models, employing different architectures for teachers and students achieves better performance than sharing the same architecture. For example, when applying a ViT-B/16 student, a ViT-L/16 teacher achieves 0.27% more accuracy than using a ViT-L/16 student. This suggests that using different architectures for teacher and student increases information diversity, which contributes to enhanced generalization for teacher models [84].

These experimental results demonstrate the effectiveness of LOT in enhancing the generalization of pretrained CNN-based and Transformer-based vision models in image classification.

3.5 Analysis of Computational Cost and Efficiency

For supervised and unsupervised tasks, LOT involves training teacher models alongside student models as outlined in Algorithm 1. Compared to Teacher-only, the potential limitation of LOT is that it requires additional computation and memory for the student models. Therefore, in our results in Section 3, we maintain the same total training steps between LOT (accumulated for the teacher and student) and Teacher-only and demonstrate that LOT achieves better generalization performance under the same number of updates. In this regard, we show the test accuracy of image classification between LOT and Teacher-only using ViT models with respect to the total training steps in Figure 3. We note that LOT achieves better test accuracy than Teacher-only in both ViT-B/16 and ViT-L/16 with fewer total training steps. Moreover, we demonstrate that LOT remains effective even when the student model is smaller than the teacher model in Table 3, which further reduces the computation cost compared to Teacher-only in the same total training steps and accommodates different student model choices with resource constraints. We provide more results regards efficiency of LOT in Appendix H.

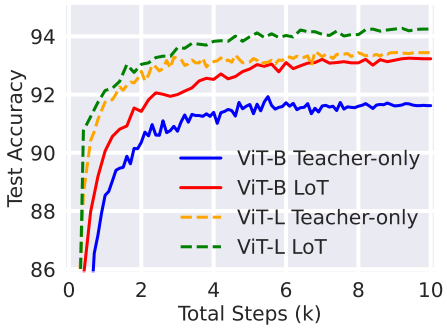

Figure 3: Test accuracy of teacher models in LOT and Teacher-only using ViT-B/16 and ViT-L/16 on CIFAR-100. LOT achieves higher test accuracy with fewer training steps.

In RL tasks, only the teacher model interacts with the environment to collect samples, and the student can learn from the teacher samples exclusively (please refer to Appendix G for the algorithm of

Table 3: The test accuracy of the teacher model for various teacher-student model combinations in LOT and the baseline. Results are averaged over three runs. LOT consistently enhances test performance in all model choices and datasets.

Pretrained	Downstream	Teacher	Student	Image Size	Teacher/Student #Param.	Teacher-only	LOT
ImageNet-1K	CIFAR-100	ResNet-18	MobileNetV2	224^2	12M / 4M	81.14 ± 0.58	**82.78** ± 0.36
		ResNet-18	ResNet-18	224^2	12M / 12M	81.14 ± 0.58	**82.89** ± 0.25
		ResNet-18	ResNet-50	224^2	12M / 26M	81.14 ± 0.58	**83.13** ± 0.26
		ResNet-50	MobileNetV2	224^2	26M / 4M	84.09 ± 0.32	**85.38** ± 0.44
		ResNet-50	ResNet-18	224^2	26M / 12M	84.09 ± 0.32	**85.77** ± 0.19
		ResNet-50	ResNet-50	224^2	26M / 26M	84.09 ± 0.32	**86.04** ± 0.38
ImageNet-21K	CIFAR-100	ViT-B/16	ViT-B/16	384^2	86M / 86M	91.57 ± 0.31	**93.17** ± 0.35
		ViT-B/16	ViT-L/16	384^2	86M / 307M	91.57 ± 0.31	**93.25** ± 0.44
		ViT-L/16	ViT-B/16	384^2	307M / 86M	93.44 ± 0.28	**94.29** ± 0.33
		ViT-L/16	ViT-L/16	384^2	307M / 307M	93.44 ± 0.28	**94.18** ± 0.26
ImageNet-21K	ImageNet-1K	ViT-B/16	ViT-B/16	384^2	86M / 86M	83.97 ± 0.11	**84.54** ± 0.15
		ViT-B/16	ViT-L/16	384^2	86M / 307M	83.97 ± 0.11	**84.80** ± 0.08
		ViT-L/16	ViT-B/16	384^2	307M / 86M	85.15 ± 0.17	**85.92** ± 0.09
		ViT-L/16	ViT-L/16	384^2	307M / 307M	85.15 ± 0.17	**85.65** ± 0.11
		Swin-B	Swin-B	384^2	88M / 88M	86.37 ± 0.06	**86.68** ± 0.15
		Swin-B	Swin-L	384^2	88M / 197M	86.37 ± 0.06	**86.73** ± 0.14
		Swin-L	Swin-B	384^2	197M / 88M	87.27 ± 0.11	**87.64** ± 0.12
		Swin-L	Swin-L	384^2	197M / 197M	87.27 ± 0.11	**87.59** ± 0.09

Table 4: Performance comparison of Teacher-only, BAN and LOT on CIFAR-100. LOT achieves superior performance to Teacher-only and BAN.

Dataset	Teacher	Student	Teacher-only	BAN (Student)	LOT (Teacher)
CIFAR-100	ResNet-18	ResNet-18	81.14	82.08	**82.89**
CIFAR-100	ResNet-50	ResNet-50	84.09	84.73	**86.04**
CIFAR-100	ViT-B/16	ViT-B/16	91.57	92.44	**93.17**
CIFAR-100	ViT-L/16	ViT-L/16	93.44	93.82	**94.18**

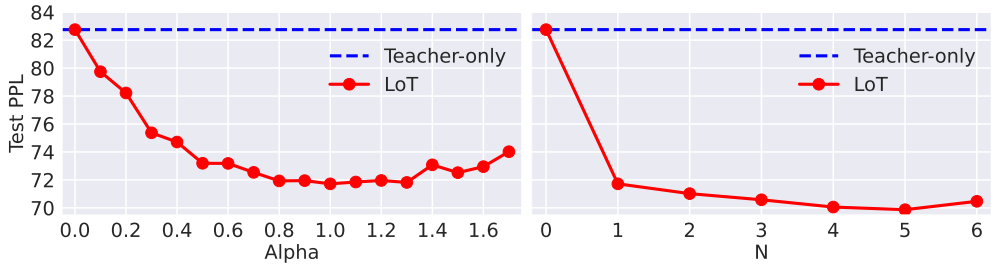

Figure 4: Effects of regularization coefficient α (left) and student steps ratio N (right). $\alpha = 1$ is the best α value to achieve the lowest test perplexity of the teacher model, and moderate student steps ratio N such as 4 and 5 benefit the teacher model the most.

PPO-version LOT). Therefore, LOT introduces negligible computation costs since sample collections are more resource-intensive than fitting the agent network to the samples in RL. For instance, in our Atari games experiments, the training time of LOT (606 minutes) is comparable to the Teacher-only setting (597 minutes) on a single NVIDIA A6000 GPU.

3.6 Additional Investigation

Comparison to KD. To investigate the effect of LOT compared to other student-teacher learning paradigms, we compare LOT to the born-again networks (BAN) baseline [29]. In BAN, we select the checkpoint with the best performance of the Teacher-only model as the (frozen) teacher and distill its knowledge into a student model with an identical architecture. Equal weights are assigned to the hard loss (from the dataset) and soft loss (from the teacher) to train the student model [36]. All other configurations remain consistent with LOT . The results in Table 4 indicate that LOT achieves superior performance than BAN with a strong feedback model, further indicating the significance of the interactive learning process in LOT .

Effect of regularization coefficient α. The strength of regularization plays a crucial role in the overall training effect [50]. To investigate the effects of LOT on the generalization of the teacher model, we perform experiments on PTB using the LSTM architecture for both teacher and student models. The configuration follows Section 3.3, except that we gradually increase the value of α in LOT from 0 to 1.7 and examine the test PPL of the teacher model. The results are presented in Figure 4 (left). We observe that the performance of the teacher model improves rapidly as α increases from 0 to 1, and when the value exceeds this point, the performance of the teacher begins to decline. This observation suggests that moderate feedback from the student is most beneficial for the teacher, but an excessively strong signal can hinder the teacher's learning process. Similar effects of large α values have been noted in joint teacher-student training in knowledge distillation [75].

Effect of student steps ratio N. To demonstrate the importance of the student steps ratio N in LOT , we conduct additional experiments by training LSTM teacher and student models on PTB using various values of N. The empirical results presented in Figure 4 (right) indicate that the teacher benefits most from a moderate N value, such as 4 or 5. This finding suggests that achieving a balanced ratio between teacher and student model updates is crucial for optimal performance. When N is too low, the student may not sufficiently learn from the teacher, thereby reducing the quality of the feedback it provides. Conversely, if N is too high, the student may overfit the teacher's errors, resulting in less effective imitability measurement.

4 Conclusion

Identifying generalizable multiscale correlations from the vast space of possible correlations remains a significant challenge in machine learning. Inspired by cognitive science beliefs about human intelligence, we have shown experimentally that generalizable correlations are more imitable by other learners. In particular, we introduced a novel regularization method, LOT , which identifies generalizable correlations by teaching student models and exploiting their feedback. We conducted comprehensive experiments across various learning tasks and neural architectures. The results demonstrate that our proposed regularizer enhances model performance effectively and efficiently. In conclusion, our proposed LOT regularization offers a promising new approach to improve the generalization of neural networks by leveraging the learning process of student models and incorporating their feedback to refine the teacher model.

5 Acknowledgments

Metaxas is partially supported by research grants from NSF: 2310966, 2235405, 2212301, 2003874, 1951890, AFOSR 23RT0630, and NIH 2R01HL127661.

References

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

[2] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arpit17a.html`.

[3] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

[4] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. *Advances in neural information processing systems*, 26, 2013.

[5] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HkezXnA9YX`.

[6] Peter Bartlett. For valid generalization the size of the weights is more important than the size of the network. *Advances in neural information processing systems*, 9, 1996.

[7] Eric Baum and David Haussler. What size net gives valid generalization? *Advances in neural information processing systems*, 1, 1988.

[8] Leo Breiman and Nong Shang. Born again trees. *University of California, Berkeley, Berkeley, CA, Technical Report*, 1(2):4, 1996.

[9] 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.

[10] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[11] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.407. URL https://aclanthology.org/2020.acl-main.407.

[12] Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AUGBfDIV9rL.

[13] Satrajit Chatterjee. Learning and memorization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 755–763. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/chatterjee18a.html.

[14] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019 (12):124018, 2019.

[15] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5008–5017, 2021.

[16] Minsu Cho, Ameya Joshi, and Chinmay Hegde. Espn: Extremely sparse pruned networks. In *2021 IEEE Data Science and Learning Workshop (DSLW)*, pages 1–8. IEEE, 2021.

[17] Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Multi-agent compositional communication learning from raw visual input. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rknt2Be0-.

[18] Sahil Chopra, Michael Henry Tessler, and Noah D Goodman. The first crank of the cultural ratchet: Learning and transmitting concepts through language. In *CogSci*, pages 226–232, 2019.

[19] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[20] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 109–116, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

[21] I. Csiszar. I-Divergence Geometry of Probability Distributions and Minimization Problems. *The Annals of Probability*, 3(1):146 – 158, 1975. doi: 10.1214/aop/1176996454. URL https://doi.org/10.1214/aop/1176996454.

[22] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[25] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021.

[26] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[27] Nicholas Frosst and Geoffrey E. Hinton. Distilling a neural network into a soft decision tree. In Tarek R. Besold and Oliver Kutz, editors, *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, volume 2071 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. URL `https://ceur-ws.org/Vol2071/CExAIIA_2017_paper_3.pdf`.

[28] Tommaso Furlanello, Jiaping Zhao, Andrew M Saxe, Laurent Itti, and Bosco S Tjan. Active long term memory networks. *arXiv preprint arXiv:1606.02355*, 2016.

[29] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/furlanello18a.html`.

[30] Lukas Galke, Yoav Ram, and Limor Raviv. What makes a language easy to deep-learn? *arXiv preprint arXiv:2302.12239*, 2023.

[31] Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning. *arXiv preprint arXiv:2304.05366*, 2023.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[33] David Haussler. Quantifying inductive bias: Ai learning algorithms and valiant's learning framework. *Artificial intelligence*, 36(2):177–221, 1988.

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

[35] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[36] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`.

[37] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[40] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research*, 22(1):10882–11005, 2021.

[41] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[42] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274): 1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

[44] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

[45] Simon Kirby. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5 (2):102–110, 2001.

[46] Simon Kirby, Hannah Cornish, and Kenny Smith. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686, 2008.

[47] Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014.

[48] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1321. URL https://aclanthology.org/D17-1321.

[49] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[51] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[52] Alex Labach, Hojjat Salehinejad, and Shahrokh Valaee. Survey of dropout methods for deep neural networks. *arXiv preprint arXiv:1904.13310*, 2019.

[53] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk8N3Sclg.

[54] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

[55] David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008.

[56] Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. *Advances in neural information processing systems*, 32, 2019.

[57] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJqFGTslg.

[58] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

[59] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081.

[60] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

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

[62] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

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

[64] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004.

[65] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

[66] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations*, 2018.

[67] Poorya Mianjy and Raman Arora. On convergence and generalization of dropout training. *Advances in Neural Information Processing Systems*, 33:21151–21161, 2020.

[68] Charles A Micchelli, Massimiliano Pontil, and Peter Bartlett. Learning the kernel function via regularization. *Journal of machine learning research*, 6(7), 2005.

[69] Ryszard S Michalski. A theory and methodology of inductive learning. In *Machine learning*, pages 83–134. Elsevier, 1983.

[70] Tom M Mitchell. The need for biases in learning generalizations. 1980.

[71] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=guetrIHLFGI.

[72] Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28, 2015.

[73] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[74] Xuefei Ning, Zifu Wang, Shiyao Li, Zinan Lin, Peiran Yao, Tianyu Fu, Matthew B Blaschko, Guohao Dai, Huazhong Yang, and Yu Wang. Can llms learn by teaching? a preliminary study. *arXiv preprint arXiv:2406.14629*, 2024.

[75] Dae Young Park, Moon-Hyun Cha, Changwook Jeong, Daesin Kim, and Bohyung Han. Learning student-friendly teacher networks for knowledge distillation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=0xs40KGnsq3.

[76] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. *Advances in neural information processing systems*, 23, 2010.

[77] Ben Poole, Jascha Sohl-Dickstein, and Surya Ganguli. Analyzing noise in autoencoders and deep networks. *arXiv preprint arXiv:1406.1831*, 2014.

[78] Yi Ren, Samuel Lavoie, Michael Galkin, Danica J Sutherland, and Aaron C Courville. Improving compositional generalization using iterated learning and simplicial embeddings. *Advances in Neural Information Processing Systems*, 36, 2024.

[79] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

[80] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

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

[82] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 515–521, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568.

[83] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[84] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.

[85] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[86] Kenny Smith, Simon Kirby, and Henry Brighton. Iterated learning: A framework for the emergence of language. *Artificial life*, 9(4):371–386, 2003.

[87] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[88] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

[89] Ilya Sutskever. An observation on generalization. `https://simons.berkeley.edu/talks/ilya-sutskever-openai-2023-08-14`, 2023.

[90] Ilya Sutskever. An observation on generalization. *Large Language Models and Transformers Workshop*, 2023. URL `https://simons.berkeley.edu/talks/ilya-sutskever-openai-2023-08-14`.

[91] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.

[92] R Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385—395, February 1997. ISSN 0277-6715.

[93] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[94] Michael Tomasello. The human adaptation for culture. *Annual review of anthropology*, 28(1): 509–529, 1999.

[95] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers via distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

[96] 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.

[97] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[98] Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. Iterated learning for emergent systematicity in vqa. In *International Conference on Learning Representations*.

[99] Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. Iterated learning for emergent systematicity in {vqa}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Pd_oMxH8IlF.

[100] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/wan13.html.

[101] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[102] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[103] Chuanguang Yang, Zhulin An, Helong Zhou, Linhang Cai, Xiang Zhi, Jiwen Wu, Yongjun Xu, and Qian Zhang. Mixskd: Self-knowledge distillation from mixup for image recognition. In *European Conference on Computer Vision*, pages 534–551. Springer, 2022.

[104] Kai Yu, Wei Xu, and Yihong Gong. Deep learning with kernel regularization for visual recognition. *Advances in neural information processing systems*, 21, 2008.

[105] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

[106] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[107] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

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

[109] Zhenzhu Zheng and Xi Peng. Self-guidance: Improve deep neural network generalization via knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3203–3212, January 2022.

A Ethics and Social Impacts

In this work, we propose a regularization method to improve the generalization of deep neural networks. Our work focuses on technical contributions to deep learning and AI. Therefore, the potential social impacts of AI in general apply to our work, including fake information, toxic content, fairness concerns, and misuse of AI. For example, toxic content like hate speech can lead to data contamination and therefore have harmful impacts on society, which has been observed in large-scale pretrained models. By employing our method, such harmful behavior can potentially be amplified.

B Related Works

B.1 Regularization in Deep Learning

Regularization serves as a primary strategy to improve generalization capabilities and mitigate over-fitting [52]. Various regularization techniques exist for deep neural networks. One of the earliest and most straightforward approaches to regularization involves constraining the model's capacity by adding a penalty function to the original objective function. Techniques such as L1 regularization [41, 92, 93], L2 regularization [20, 50, 82], and weight decay [50, 104] fall into this category. Introducing noise [38, 77] to the system can also judiciously enhance generalizability and prevent over-fitting. Dropout [4, 37, 88, 100] is a widely used regularization technique that randomly drops certain neural network connections during training.

B.2 Student-Teacher Learning Paradigms

B.2.1 Knowledge Distillation

Knowledge distillation (KD) is a technique that transfers knowledge from a teacher model to a student model by training the student to imitate the teacher's outputs [36]. This approach is widely applied in areas such as model compression, transparency, and interpretability [8, 10, 27, 36, 60, 91]. Model compression is often motivated by resource constraints. Pioneering works include Buciluǎ et al. [10], which compresses ensemble neural networks into a single network, and Ba and Caruana [3], which improves shallow neural network accuracy by mimicking deep networks. KD is also applied in various domains, including deep reinforcement learning [80], continual learning [28, 59, 85], and learning privileged information theory [62, 76]. The dark knowledge method [36] further develops KD, where a student model aims to fully match the output distribution of the teacher. Intuitively, distillation is effective because the teacher's output distribution over classes provides a more informative training signal than a one-hot label. Additionally, in born-again networks (BAN) [29], the teacher and student have identical neural architecture and model sizes, but the student can surprisingly surpass the teacher's accuracy.

B.2.2 Language Emergence

In a cooperative environment, agents can learn emergent languages for communication to solve specific tasks. The emergence of such communication protocols is extensively studied in the context of multi-agent referential games [26, 55]. In these games, one agent is required to describe its observations to another agent, which is then tasked with deducing the initial agent's observations [53]. The majority of methods employed to learn discrete communication protocols between agents utilize RL [26, 99]. Compositionality is a desirable feature in the language used by agents, as it facilitates flawless generalization for previously unseen combinations of attributes [5, 11, 17, 48]. However, the community still lacks strong research indicating what general conditions are necessary or sufficient for compositional language emergence. Chaabouni et al. [11, 12], Galke et al. [30], Li and Bowling [56] postulate that compositional languages are more straightforward to learn.

C Motivation and Insights of Our Method

The concept of Learning from Teaching originates in cognitive psychology and linguistics, particularly within the iterated learning theory of language emergence [45–47, 86]. This theory posits that the generalizable nature of languages arises from the iterative learning process across generations in a

society. The core hypothesis is that a generalizable language is inherently easier to teach and learn [56, 78, 98], which aligns with our main hypothesis.

In the AI community, recent research has aimed to employ iterated learning to enhance the generalization of emergent languages and language acquisition among artificial learners. For example, some studies have used iterated learning to improve the generalization of emergent languages between AI agents [56, 78], while others have applied it to address generalization challenges in tasks like compositional Visual Question Answering (VQA) [98]. LOT shares the same motivation as this line of research. Our primary contribution extends the concept of "ease-of-teaching" [56] from language learning to a broader range of machine learning tasks, including supervised, unsupervised learning, and reinforcement learning.

LOT functions as a regularizer, similar to other commonly used regularizers like the L2 regularizer. The L2 regularizer is effective because it encourages neural networks to learn simpler correlations, thereby avoiding overfitting. It is widely accepted that correlations with lower Kolmogorov complexity are more generalizable if they can perfectly explain a complex dataset. This aligns with the idea that "generalization equals optimal compression," as discussed by Ilya Sutskever [89]. Essentially, this notion adapts Occam's Razor to the field of AI. Our key insight is that the "ease-of-teaching" metric serves as an effective regularizer beyond language emergence tasks.

Consider an intuitive example: Student A learns math by rote memorization, while Student B understands the core concepts and only memorizes essential rules, deducing the rest when needed. Both approaches can perform similarly on simple problem sets. However, as data complexity increases, Student A's burden grows significantly, while Student B's understanding-based approach remains manageable. Consequently, Student B's knowledge is easier to teach to another student, as it involves less complexity. Therefore, teachability (or imitability) can serve as a proxy for complexity.

D Implementation Details.

Atari Games. We perform experiments on four Atari games, namely Beam-Rider, Breakout, UpNDown, and Gravitar, following the implementation outlined in [42]. We set the regularization coefficient α to 0.5 for BeamRider, Breakout, and UpNDown, and to 0.1 for Gravitar. The other hyperparameters remain consistent across all four games. We use N of 5. For all agents, the optimizer employed is Adam, with an initial learning rate of 0.00025. The teacher agent is trained for a total of $20{,}000{,}000$ timesteps. The temperature used in the KL loss is set to 1. The experiments are implemented on the NVIDIA A6000 48GB GPUs.

Language Modeling. In the training-from-scratch experiments, we use the Transformer-XL architecture following Dai et al. [22], the LSTM architecture following Zaremba et al. [106], and the AWD-LSTM architecture following Merity et al. [66]. For supervised fine-tuning experiments with LLaMA-1 and LLaMA-2, we employ the hyperparameters described in Yue et al. [105] and use the HuggingFace Transformers library [101]. The hyperparameters for LOT are detailed in Table 5. The experiments for LSTM and AWD-LSTM are implemented on one single NVIDIA A100 40GB GPU. The Transformer-XL and LLaMA of LOT are trained on 4 and 8 NVIDIA A100 40GB GPUs, respectively.

Model	Dataset	α	N	Optimizer	Learning Rate	Training Epochs/Steps	Temperature
LSTM	PTB	1.0	1	SGD	30	30 Epochs	1.5
AWD-LSTM	PTB	1.0	1	ASGD	30	250 Epochs	1.5
Transformer-XL-B	WikiText-103	0.1	1	ADAM	0.01	60,000 Steps	2
Transformer-XL-L	WikiText-103	0.1	1	ADAM	0.01	150,000 Steps	2
LLaMA-1 7B	GSM8K	0.01	1	ADAMW	2×10^{-5}	2 Epochs	2
LLaMA-1 7B	MATH	0.01	1	ADAMW	2×10^{-5}	2 Epochs	2
LLaMA-2 7B	GSM8K	0.01	1	ADAMW	2×10^{-5}	2 Epochs	2
LLaMA-2 7B	MATH	0.01	1	ADAMW	2×10^{-5}	2 Epochs	2

Table 5: Hyperparameters for Language Modeling.

Image Classification. For CNN experiments, we use the ImageNet-1K pretrained architectures MobileNetV2 and ResNets, which can be downloaded from the official PyTorch Model Zoo[2]. For

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

ViT and Swin experiments, we follow the implementations described in Dosovitskiy et al. [24] and Liu et al. [61], using the official ImageNet-1K or ImageNet-21K pretrained weights downloaded from [3] and [4]. The optimal hyperparameters for LOT are obtained through grid research. The detailed hyperparameters are illustrated in Table 6. The experiments for MobileNetV2 and ResNets are implemented on one single NVIDIA A100 40GB GPU. The ViT and Swin experiments are implemented on 4 NVIDIA A100 40GB GPUs.

Model	Dataset	α	N	Optimizer	Learning Rate	Training Epochs/Steps	Temperature
MobileNetV2	CIFAR-100	1.0	1	SGD	0.02	30 Epochs	1.5
ResNet-18	CIFAR-100	1.0	1	SGD	0.02	30 Epochs	1.5
ResNet-50	CIFAR-100	1.0	1	SGD	0.02	30 Epochs	1.5
ViT-B/16	CIFAR-100	1.0	1	SGD	0.02	5,000 Steps	1.5
ViT-L/16	CIFAR-100	1.0	1	SGD	0.02	5,000 Steps	1.5
ViT-B/16	ImageNet-1K	1.0	1	SGD	0.03	10,000 Steps	1.5
ViT-L/16	ImageNet-1K	1.0	1	SGD	0.03	10,000 Steps	1.5
Swin-B	ImageNet-1K	0.5	1	ADAMW	2×10^{-5}	15 Epochs	1.5
Swin-L	ImageNet-1K	0.5	1	ADAMW	2×10^{-5}	15 Epochs	1.5

Table 6: Hyperparameters for Image Classification.

E Scalability Analysis

From our extensive results shown in Section 3, LOT proves to be widely applicable across various domains, including reinforcement learning (Section 3.2), unsupervised learning (Section 3.3), and supervised learning (Section 3.4). It can be effectively applied to different architectures such as CNN-based (Table 3), LSTM-based (Table 1), and Transformer-based(Table 1) models. LOT works well on both small datasets like PTB (Table 1) and CIFAR-100 (Table 3), and large datasets such as WikiText-103 (Table 1) and ImageNet (Table 3). It is also suitable for both small models like ResNets (Table 3) and large models like ViT (Table 3) and LLaMA (Table 2). Additionally, LOT is compatible with existing regularization methods such as weight decay and dropout. In our experiments with ResNets, weight decay was applied to both LOT and Teacher-only setups. In the experiments with Transformer-XL, ViT, and Swin, dropout is applied to both LOT and Teacher-only setups.

F Limitation

A potential limitation of LOT lies in the additional computational and memory costs required for training the student models. However, as demonstrated in Section 3.5, LOT achieves better generalization with fewer training steps compared to Teacher-only models, and the flexibility in choosing student models can accommodate varying resource constraints. In RL, the additional computational costs introduced by LOT are negligible, as sample collection is more resource-intensive than fitting the agent networks to the samples, as discussed in Section 3.5. Moreover, in real-world settings, inference cost is more critical than training cost. The superior generalization achieved by LOT offers significant benefits during inference without introducing additional inference costs.

G Algorithm for the PPO-version of Our Method

The LOT algorithm for Proximal Policy Optimization (PPO) is illustrated in Algorithm 2. In our experiments, the teacher's sampled data $\mathcal{B}_t$ is continuously added to the student sample collections $\mathcal{D}_s$. Meanwhile, the most recent samples from $\mathcal{D}_s$ are used to formulate the student training batch $\mathcal{B}_s$ to ensure a high quality of its training dataset.

H Additional Results

Computational Efficiency. To further demonstrate the computational efficiency and superiority of LOT , we conduct experiments using LSTM on PTB and ViT-B/16 on CIFAR-100 with varying training epochs and steps, while keeping other configurations the same as in Section 3.3 and Section 3.4. The results presented in Table 7 demonstrate that given equivalent computational budgets, LOT consistently outperforms the Teacher-only model across various datasets and architectures, even when the Teacher-only model trains for twice the number of epochs and steps. This further highlights

[3] `https://github.com/google-research/vision_transformer`

[4] `https://github.com/microsoft/Swin-Transformer`

Algorithm 2 Learning from Teaching for PPO

Input: Regularization Coefficient $\alpha > 0$, Student Steps Ratio $N > 0$.
Initialize teacher network T parameterized by $\boldsymbol{\theta}$ and student networks $S_i, i = 1, 2, \cdots K$, parameterized by $\boldsymbol{\phi}$.
Initialize replay buffer $\mathcal{D}_s = \emptyset$
repeat
  Sample minibatch $\mathcal{B}_t$ by running T in simulator, add $\mathcal{B}_t$ to $\mathcal{D}_s$
  Sample a batch of data $\mathcal{B}_s \subset \mathcal{D}_s$
  Compute $\tilde{R}(\boldsymbol{\theta}) = \frac{\alpha}{|\mathcal{B}_s|} \sum_{\mathbf{x} \in \mathcal{B}_s} \sum_{i=1}^{K} \lambda_i \mu_{t,s_i}(\mathbf{x})$
  Compute $\tilde{L}_t(\boldsymbol{\theta})$ using the PPO loss on minibatch $\mathcal{B}_t$
  Update $\boldsymbol{\theta}$ using gradient $\nabla_{\boldsymbol{\theta}} \tilde{L}_t(\boldsymbol{\theta})$
  Fit value network for PPO on minibatch $\mathcal{B}_t$
  for $i = 1$ **to** N **do**
    Sample $\mathcal{B}_s \subset \mathcal{D}_s$
    Compute $\tilde{L}_s(\boldsymbol{\phi}) = \frac{1}{|\mathcal{B}_s|} \sum_{\mathbf{x} \in \mathcal{B}_s} \sum_{i=1}^{K} \mu_{s_i,t}(\mathbf{x})$
    Update student networks' parameters $\boldsymbol{\phi}$ using loss gradient $\nabla_{\boldsymbol{\phi}} \tilde{L}_s(\boldsymbol{\phi})$
  end for
until T converges

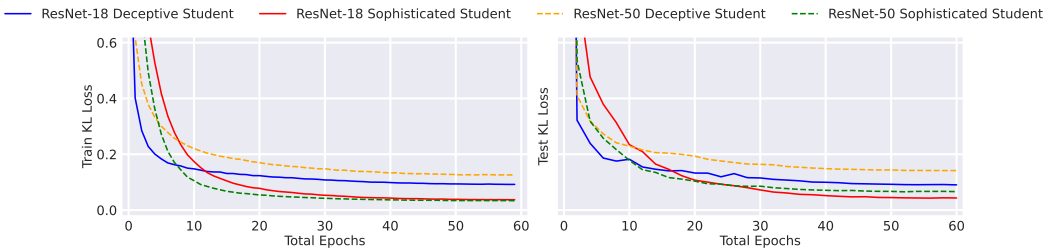

Figure 5: Training and test KL-divergence losses of student models in LOT using ResNet-18 and ResNet-50 on CIFAR-100 with different teacher models.

LOT 's effectiveness in improving the teacher model's generalization while maintaining enhanced computational efficiency.

Table 7: Performance of the teacher model in LOT and Teacher-only on image classification. The hyperparameters are the same as the corresponding experiments in the paper.

Dataset	Teacher	Student	Total Train Epochs/Steps	Teacher-only	LOT
CIFAR-100	ViT-B/16	ViT-B/16	10,000 steps	91.57	**93.17**
CIFAR-100	ViT-B/16	ViT-B/16	15,000 steps	91.74	**93.23**
CIFAR-100	ViT-B/16	ViT-B/16	20,000 steps	91.82	**93.40**
PTB	LSTM	LSTM	60 epochs	82.75	**71.72**
PTB	LSTM	LSTM	90 epochs	82.48	**71.22**
PTB	LSTM	LSTM	120 epochs	82.42	**70.67**

Additional Evidence for Hypothesis. We provide additional experimental results to validate our hypothesis using ResNet-50 and ResNet-18 as both the teacher and student models on CIFAR-100, following the same methodology described in Section 3.1, but with different model architectures. The training and test KL-divergence of the sophisticated and deceptive students are shown in Figure 5. We observe that the sophisticated students achieve lower final KL losses compared to the deceptive students with fewer training epochs, which further supports our hypothesis

Out-of-distribution Performance. We conduct additional experiments by fine-tuning models on ImageNet-1K and evaluating them on ImageNet-R and ImageNet-Sketch using ViT-B/16 and ViT-L/16 models to investigate the out-of-distribution robustness of LOT . The results, shown in

Table 8, demonstrate that LOT also brings performance improvements on these datasets, indicating the robustness of LOT across a broader set of scenarios.

Additional Comparison to KD. In Table 4, we show that LOT outperforms the distillation method BAN. To provide stronger validation of LOT 's effectiveness, we conduct additional experiments using ResNet-50 and ViT-B/16 on CIFAR-100. We compare LOT to distillation methods such as BAN, DKD [108], and ReviewKD [15], with the teacher weights in these methods being the best checkpoint of Teacher-only. The results, shown in Table 9, indicate that LOT achieves better performance than these distillation baselines, further underscoring the effectiveness of the unique interactive learning process of LOT .

Table 8: Performance of LOT and Teacher-only on ImageNet-R and ImageNet-Sketch.

Dataset	Teacher	Student	Teacher-only / LOT
ImageNet-R	ViT-B/16	ViT-B/16	49.11 / 52.27
ImageNet-R	ViT-B/16	ViT-L/16	49.11 / 54.08
ImageNet-R	ViT-L/16	ViT-B/16	54.42 / 58.18
ImageNet-R	ViT-L/16	ViT-L/16	54.42 / 57.79
ImageNet-Sketch	ViT-B/16	ViT-B/16	38.85 / 41.46
ImageNet-Sketch	ViT-B/16	ViT-L/16	38.85 / 42.89
ImageNet-Sketch	ViT-L/16	ViT-B/16	43.83 / 47.61
ImageNet-Sketch	ViT-L/16	ViT-L/16	43.83 / 45.91

Table 9: Performance of LOT , BAN, ReviewKD, DKD on CIFAR100.

Method	Teacher	Student	Accuracy
Teacher-only	ResNet-50	N/A	84.09
BAN	ResNet-50	ResNet-50	84.73
ReviewKD	ResNet-50	ResNet-50	85.31
DKD	ResNet-50	ResNet-50	85.17
LOT	ResNet-50	ResNet-50	86.04
Teacher-only	ViT-B/16	N/A	91.57
BAN	ViT-B/16	ViT-B/16	92.44
ReviewKD	ViT-B/16	ViT-B/16	92.73
DKD	ViT-B/16	ViT-B/16	92.82
LOT	ViT-B/16	ViT-B/16	93.17

Results on Validation Datasets. We provide additional results on the official validation datasets for PTB and WikiText-103 in Table 10. These results demonstrate that LOT consistently outperforms the Teacher-only approach on both the validation and test datasets for PTB and WikiText-103, further validating the effectiveness of LOT .

Table 10: Test/Validation perplexity of LOT and Teacher-only on the official test/validation datasets.

Dataset	Teacher	Student	Teacher-only (Valid)	Teacher-only (Test)	LOT (Valid)	LOT (Test)
PTB	LSTM	LSTM	86.02	82.75	73.98	71.72
PTB	AWD-LSTM	AWD-LSTM	60.62	58.69	55.07	53.31
Wikitext-103	Transformer-XL-B	Transformer-XL-B	24.68	23.72	22.24	21.65
Wikitext-103	Transformer-XL-L	Transformer-XL-L	18.65	18.50	16.41	16.47

Performance of Student Models. We present the results for the student models in Table 11. Our observations indicate that when the student and teacher models share the same architecture, the student models can achieve performance levels comparable to those of the teacher models. While the performance of the student models improves under LOT , it is important to highlight that LOT is primarily designed to enhance the generalization capabilities of the teacher model.

Detialed Computation Cost. We provide a detailed comparison of the computational budget for LOT and Teacher-only in Table 12. Our analysis shows that LOT uses the same number of CPU cores as Teacher-only, with GPU usage being 12% to 55% higher. Despite this, LOT exhibits lower training times compared to Teacher-only (except in RL tasks) when subjected to the same total training epochs/steps, while still achieving significant performance improvements.

Table 11: The performance of student models in LoT on language modeling and image classification.

Task	Dataset	Teacher	Student	Teacher-only	LOT (Teacher)	LOT (Student)
Language Modeling	PTB	LSTM	LSTM	82.75	71.72	73.33
Language Modeling	WikiText-103	Transformer-XL-L	Transformer-XL-L	18.50	16.47	16.89
Image Classification	CIFAR100	ResNet-50	ResNet-18	84.09	85.77	83.24
Image Classification	CIFAR100	ResNet-50	ResNet-50	84.09	86.04	85.72
Image Classification	ImageNet-1K	ViT-B/16	ViT-B/16	91.57	93.17	92.95
Image Classification	ImageNet-1K	ViT-B/16	ViT-L/16	91.57	93.25	93.89

Table 12: Computational resources, memory usage, and training time of LOT and Teacher-only.

Dataset	Teacher Model / Student Model	Total Train Steps (teacher+student)	Computational Resources	CPU Usage (Teacher-only/LOT)	GPU Usage (Teacher-only/LoT)	Training Time (Teacher-only/LOT)	Performance (Teacher-only/LOT)
BeamRider	Standard Network / Standard Network	20M frames	1 NVIDIA A6000 48GB GPU	16 core / 16 core	0.8 GB / 0.9 GB	10 h / 10.1 h	3,651 score / 5,956 score (↑)
PTB	LSTM / LSTM	60 epochs	1 × NVIDIA A100 40GB GPU	1 core / 1 core	1.1 GB / 1.5 GB	0.6 h / 0.3 h	82.8 ppl / 71.7 ppl (↓)
WikiText-103	Transformer-XL-L / Transformer-XL-L	0.3M steps	4 × NVIDIA A100 40GB GPU	4 core / 4 core	4 × 21.4 GB / 4 × 33.2 GB	85.6 h / 67.7 h	18.5 ppl / 16.5 ppl (↓)
GSM8K	LLaMA-2 7B / LLaMA-2 7B	4 epochs	8 × NVIDIA A100 40GB GPU	8 core / 8 core	8 × 27.4 GB / 8 × 39.8 GB	8.1 h / 6.7 h	39.8 acc / 41.9 acc (↑)
CIFAR100	ResNet-50 / ResNet-18	60 epochs	1 × NVIDIA A100 40GB GPU	1 core / 1 core	13.6 GB / 16.7 GB	0.7 h / 0.5 h	84.1 acc / 85.8 acc (↑)
ImageNet-1K	ViT-L/16 / ViT-B/16	20K steps	4 × NVIDIA A100 40GB GPU	4 core / 4 core	4 × 17.5 GB / 4 × 23.1 GB	28.9 h / 18.7 h	85.2 acc / 86.0 acc (↑)

Ablation of Metrics in LOT Regularizer. We conduct experiments with different metrics for the "imitability" measurement, such as L2 loss. However, we find that using KL-divergence achieves better performance compared to L2 loss. The results of utilizing L2 loss for the LOT regularizer with ViT-B/16 and ViT-L/16 on CIFAR-100 are presented in Table 13. These results show that using L2 loss for the LOT regularizer also brings performance improvements, further indicating the effectiveness of LOT regularization.

Table 13: Performance of using L2 loss for the LOT regularizer on CIFAR100.

Dataset	Teacher	Student	Teacher-only	LOT (KL-Divergence)	LOT (L2)
CIFAR100	ViT-B/16	ViT-B/16	91.57	93.17	92.77
CIFAR100	ViT-B/16	ViT-L/16	91.57	93.25	92.94
CIFAR100	ViT-L/16	ViT-B/16	93.44	94.29	94.12
CIFAR100	ViT-L/16	ViT-L/16	93.44	94.18	94.05

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction (Section 1) in this paper reflect the contributions of our method. The strong results in our experiments further reflect our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The potential limitation of LoT lies in the additional computational and memory costs required for training the student models. However, we demonstrate that LoT achieves better generalization with fewer training steps compared to Teacher-only models and the flexibility in choosing student models can accommodate varying resource constraints. We provide discussions in Section 3.5 and Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The implementation details of this paper is fully illustrated in Appendix D and we make an extensive effort to ensure the reproducibility of the results in this paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code of this paper in the additional supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting and details are introduced in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All our main results are averaged over multiple runs and the error bar are provided in our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources details are provided in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impacts of this paper is discussed in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not release new models or datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly popular datasets and models and obtain the license of using LLaMA models. We credit the license and term of use in the code in the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include license in the code in the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.