SEEKING NEURAL NUGGETS: KNOWLEDGE TRANSFER IN LARGE LANGUAGE MODELS FROM A PARAMETRIC PERSPECTIVE

Ming Zhong[1], **Chenxin An**[2], **Weizhu Chen**[3], **Jiawei Han**[1], **Pengcheng He**[3]
[1]University of Illinois Urbana-Champaign, [2]The University of Hong Kong, [3]Microsoft Azure AI
{mingz5, hanj}@illinois.edu, cxan23@connect.hku.hk
wzchen@microsoft.com, Herbert.he@gmail.com

ABSTRACT

Large Language Models (LLMs) inherently encode a wealth of knowledge within their parameters through pre-training on extensive corpora. While prior research has delved into operations on these parameters to manipulate the underlying implicit knowledge — encompassing detection, editing, and merging — there remains an ambiguous understanding regarding their transferability across models with varying scales. In this paper, we seek to empirically investigate knowledge transfer from larger to smaller models through a parametric perspective. To achieve this, we employ sensitivity-based techniques to extract and align knowledge-specific parameters between different LLMs. Moreover, the LoRA module is used as the intermediary mechanism for injecting the extracted knowledge into smaller models. Evaluations across four benchmarks validate the efficacy of our proposed method. Our findings highlight the critical factors contributing to the process of parametric knowledge transfer, underscoring the transferability of model parameters across LLMs of different scales. Project website: https://maszhongming.github.io/ParaKnowTransfer.

1 INTRODUCTION

Driven by the advancements of Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023; Touvron et al., 2023a), a transformative wave has reshaped the landscape in multiple areas of Artificial Intelligence, elevating performance across diverse tasks. From a parametric perspective, the objective of pre-training is to encode substantial amounts of knowledge into model parameters through language modeling on extensive corpora (Peters et al., 2018; Radford et al.; Devlin et al., 2019; Delétang et al., 2023). In a quest to unravel the intricate workings of LLMs, a multitude of research efforts have been directed toward the detailed exploration and nuanced manipulation of this reservoir of implicit knowledge.

Early research efforts sought to detect this parametric knowledge, typically probing the concrete facts by using the "fill-in-the-blank" task under a closed-book setting (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020). Subsequent studies delved into the feasibility of executing operations on model knowledge, including knowledge editing (Cao et al., 2021; Mitchell et al., 2022; Meng et al., 2022), a technique designed to modify targeted knowledge while preserving the integrity of the remaining information, and model merging (Izmailov et al., 2018; Ainsworth et al., 2023; Stoica et al., 2023), a strategy that combines multiple models to enhance robustness or facilitate multitasking abilities. While these investigations exhibited that such parametric knowledge is both *detectable* and *editable* within a single model, the broader question of whether it is *transferable* across different LLMs remains an open and under-explored topic.

Knowledge transfer refers to distilling the expertise of larger teacher models into smaller, more manageable counterparts, thereby democratizing access to cutting-edge machine learning capabilities. As illustrated in Figure 1, online and offline distillation currently stand as the primary paradigms. The former, especially prevalent before the LLM era, capitalizes on teacher models to guide the learning trajectory of student models (Hinton et al., 2015; Sanh et al., 2019; Gou et al., 2021). Yet, as LLMs grow in scale, the inherent demand for the teacher model to undergo fine-tuning or participate in student training becomes increasingly cost-prohibitive. In contrast, offline distillation

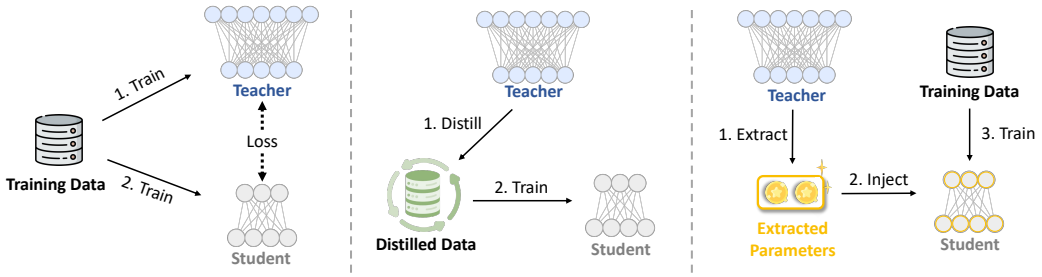


Figure 1: Different paradigms of knowledge transfer from teacher models to student models. (a) Online Distillation: utilizing soft logits from the fine-tuned teacher model to guide the training of the student model; (b) Offline Distillation: generating a distilled dataset that encapsulates the knowledge of the teacher model to fine-tune the student model. (c) Parametric Knowledge Transfer: extracting knowledge-specific parameters from the vanilla teacher model and injecting them into the student model to enhance its training efficacy.

calls upon the teacher model merely to generate answers to open-ended queries, creating a distilled training dataset for students (Honovich et al., 2023; Wang et al., 2023c; Taori et al., 2023). Despite reducing the overhead to thousands of inferences, it completely overlooks the rich knowledge implicitly stored within the teacher's parameters.

In this paper, we empirically investigate knowledge transfer from a distinct parametric perspective, dedicated to selecting static parameters directly from the teacher model and exploring their transferability. Specifically, we introduce a new parametric knowledge transfer paradigm designed to extract task-specific parameters from the teacher model and subsequently inject them into the student model, thereby enhancing performance on downstream tasks. Through decoding on a limited set of seed samples (e.g., 32 samples) with the teacher model, we calculate sensitivity metrics that serve as the basis for knowledge extraction. Given the discrepancies in the number of layers and dimensions across varied LLM scales, we employ sensitivity-based layer mapping and dimensionality reduction techniques to establish alignment between the teacher and student models. Lastly, we leverage LoRA (Hu et al., 2022) as a bridge to inject these extracted parameters into student models, facilitating its fine-tuning on downstream tasks and thus achieving the knowledge transfer process.

Experimentally, we evaluate the parametric knowledge transfer framework across four benchmark categories: reasoning (Cobbe et al., 2021), professional knowledge (Hendrycks et al., 2021), instruction-driven NLP tasks (Wang et al., 2022), and open-ended conversation (Dubois et al., 2023), using various sizes of LLaMA models (Touvron et al., 2023a;b). The results indicate that upon transferring the teacher model's parameters, the student performance demonstrates consistent improvements across all benchmarks, affirming the transferability of parametric knowledge. Furthermore, our detailed analyses illustrate the underlying factors that contribute to effective parametric knowledge transfer, discovering that the teacher scales, initialization strategies, number of seed samples, and the origin and structure of the extracted parameters all play crucial roles.

To summarize, the key contributions of this paper are threefold: (1) From a distinct perspective, we introduce a parametric knowledge transfer paradigm that encompasses stages of sensitivity-based knowledge extraction and LoRA-driven knowledge injection. (2) Through comprehensive evaluations, we provide empirical evidence that implicit model knowledge is indeed *transferable* across varying scales of LLMs. (3) Further enriching our insights into parametric knowledge transfer, we undertake a thorough analysis to pinpoint the pivotal factors that dictate its efficacy.

2 Related Work

2.1 Manipulation of Implicit Model Knowledge

With the recognition of the vast repository of knowledge embedded in model parameters (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020; Dai et al., 2022), ensuing research has sought to execute diverse operations on these parameters, aiming to manipulate the implicit knowledge.

For instance, knowledge editing endeavors to modify or update specific facts by editing the associated parameters, all the while ensuring the broader knowledge base remains untouched (Cao et al., 2021; Mitchell et al., 2022; Meng et al., 2022; 2023; Yao et al., 2023b). Another avenue, model merging and composition, combines the weights of two or more models into a unified weight set or dynamically activates different modules for greater robustness and multitasking capabilities (Huang et al., 2017; Izmailov et al., 2018; Ainsworth et al., 2023; Stoica et al., 2023; Zhong et al., 2024). Additionally, a strand of studies probes into performing arithmetic operations on the pre-trained weights, thus enabling the model to augment or diminish particular functionalities (Ilharco et al., 2023; Ortiz-Jiménez et al., 2023; Zhang et al., 2023). However, these explorations are limited to operations within individual models or merging models with identical architectures, without investigating if implicit knowledge between different scale models can be manipulated and transferred.

2.2 Inheritance of Model Knowledge

Another line of research that aligns more closely with our work concerns the operation of model parameters across scales, specifically the concept of model growth (Yao et al., 2023a; Li et al., 2023). This refers to accelerating the pre-training of LLMs by incrementally growing and expanding the parameters of a smaller model, using them as an initialization for the larger one. The majority of existing work is concentrated on devising function-preserving methods (Chen et al., 2016), ensuring that the initialized larger model replicates the behaviors of the original smaller model (Wei et al., 2016; Gu et al., 2021; Chen et al., 2022; Evci et al., 2022; Shen et al., 2022; Gesmundo & Maile, 2023). Concurrently, several studies adopt data-driven strategies, investigating reverse distillation (Qin et al., 2022) or mapping learned weights from smaller models to their larger counterparts (Wang et al., 2023a). In contrast to this research direction, our emphasis is on the transfer of knowledge from larger teacher to smaller student models, with the aim of exploring not only the efficiency of training, but also the transferability of parametric knowledge across different scenarios.

2.3 Transfer of Model Knowledge

Knowledge transfer is a research area dedicated to training a smaller student model to mimic the behavior of a larger pre-trained teacher model to achieve similar performance with fewer parameters (Hinton et al., 2015). Despite progress in improving the online distillation paradigm (Zhang et al., 2018; Lan et al., 2018; Jin et al., 2019; Mirzadeh et al., 2020; Park et al., 2021; Pham et al., 2021; Zhou et al., 2022) and optimizing the efficiency of offline distillation (Honovich et al., 2023; Wang et al., 2023c; Wu et al., 2023; Taori et al., 2023; Peng et al., 2023; Xu et al., 2023), they both completely ignore the implicit knowledge embedded inherently in the teacher model. Concurrently, Xu et al. (2024) propose weight selection for uniformly selecting parameters from a larger teacher model to initialize a smaller variant. In contrast to our work, they concentrate on vision tasks and aim to broadly enhance the capabilities of student models, rather than seeking to affirm the transferability of task-specific implicit knowledge between various models.

3 Parametric Knowledge Transfer

In this section, we first outline the task formulation for parametric knowledge transfer. Following this, we delve into a detailed description of our proposed method, as depicted in Figure 2.

3.1 Task Formulation

The core objective of parametric knowledge transfer is to enhance a student model by selectively transferring task-specific parametric knowledge from a more knowledgeable teacher model. Given a task $\mathcal{T}$, the transfer process begins with a teacher model M_T endowed with parameters $\mathbf{\Theta}_T$ and a student model M_S characterized by parameters $\mathbf{\Theta}_S$.

The first step in this procedure involves extraction, where task-relevant parameters are identified from the teacher model and resized to a desired scale based on the student model's parameter dimensions. This can be expressed as:

$$\text{Extract}(\mathbf{\Theta}_T; \mathbf{\Theta}_S; \mathcal{T}) = \mathbf{\Theta}_{T_{\text{extract}}}, \tag{1}$$

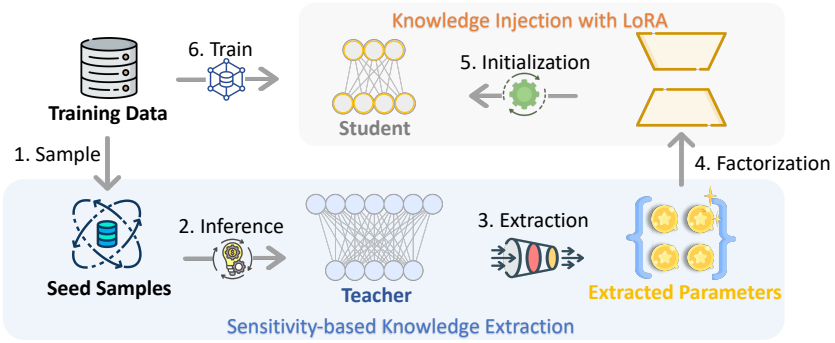


Figure 2: Overview of our parametric knowledge transfer framework. Starting with the teacher model, we compute sensitivity metrics using a set of seed samples, which aids in the extraction of task-specific knowledge. Subsequently, the extracted parameter matrices are factorized to initialize the student model's LoRA module, serving as a bridge for knowledge injection.

with Extract$(\cdot)$ encapsulating the logic for both parameter extraction and downscaling. Following extraction, the next step is the injection of these extracted parameters into the student model:

$$\text{Inject}(\mathbf{\Theta}_S; \mathbf{\Theta}_{T_{\text{extract}}}) = \mathbf{\Theta'}_S, \tag{2}$$

yielding a student model now characterized by the modified parameter set $\mathbf{\Theta'}_S$. Upon completing the knowledge injection, there remains an optional phase wherein the student model fine-tunes the newly incorporated parameters $\mathbf{\Theta'}_S$ with respect to the task $\mathcal{T}$.

3.2 Knowledge Extraction

In implementing our Extract$(\cdot)$ function, we employ parameter sensitivity as the foundational metric to guide the extraction process.

Sensitivity of the Parameters. Parameter sensitivity serves as a mechanism to measure the variation in the loss upon setting a particular parameter to zero (Mozer & Smolensky, 1988; Lee et al., 2019; Lubana & Dick, 2021; Liang et al., 2022). When this removal elicits a significant shift in the loss, such a parameter is deemed to be of high sensitivity. For a teacher model M_T with parameters $\mathbf{\Theta}_T = [\theta_1, \ldots, \theta_{N_T}]$, where N_T represents the total number of parameters, the i-th parameter can be expressed as $\mathbf{\Theta}_{T_i} = [0, \ldots, \theta_i, \ldots, 0]$. With gradients of the loss relative to $\mathbf{\Theta}_T$ represented as $\nabla_{\mathbf{\Theta}_T}\mathcal{L}$, the sensitivity of the i-th parameter for a specific sample x_j from task $\mathcal{T}$ is determined as:

$$S_{i,j} = \left| \mathbf{\Theta}_{T_i}^\top \nabla_{\mathbf{\Theta}_T} \mathcal{L}(x_j) \right|. \tag{3}$$

The rationale behind this sensitivity definition stems from the first-order Taylor expansion of $\mathcal{L}(\cdot)$ relative to θ_i at $\mathbf{\Theta}_{T_i}$ (Molchanov et al., 2017). In essence, $S_{i,j}$ provides an approximation for how the loss might change in the absence of θ_i:

$$\mathbf{\Theta}_{T_i}^\top \nabla_{\mathbf{\Theta}_T} \mathcal{L}(x_j) \approx \mathcal{L}(\mathbf{\Theta}_T) - \mathcal{L}(\mathbf{\Theta}_T - \mathbf{\Theta}_{T_i}). \tag{4}$$

To ascertain S_i for parameter i pertaining to task $\mathcal{T}$, we randomly sample k instances as seed samples for an efficient and representative estimate. Thus, the final formulation S_i for task $\mathcal{T}$ integrates the cumulative sensitivity over the sampled instances, calculated as $S_i = \sum_{j=1}^{k} S_{i,j}$.

Layer Selection and Dimensionality Reduction. Given that models of varying scales often differ in both the number of layers and their dimensions, we adopt a method of layer selection and dimensionality reduction based on sensitivity scores. Our first step is to assess the layers of the teacher model, M_T, with respect to their relevance to task $\mathcal{T}$. Let L_T and L_S represent the total number of layers in the teacher and student models, respectively, with $L_S \leq L_T$. For each layer l in M_T, we calculate a layer-specific sensitivity score, S_{T_l}, by aggregating the sensitivity scores of all parameters within that layer, represented as: $S_{T_l} = \sum_{\theta_i \in \mathbf{\Theta}_{T_l}} S_i$. Having computed the sensitivity scores

for layer l in M_T, we proceed to arrange them in descending order and select the top L_S layers with the highest scores. To preserve the inherent structure of the teacher model, the chosen layers are subsequently mapped to the student model maintaining their original sequential order.

Upon alignment of the layers, the parameter dimensions of each layer in the teacher model typically continue to surpass those of the student model. During this phase of the transfer process, our focus is primarily on all the two-dimensional matrices in the teacher model, which are denoted as $\mathbf{W_T}$. We initially identify submatrices within each $\boldsymbol{W}_{T_i} \in \mathbf{W_T}$ that match the student model's matrix dimensions, $\mathbb{R}^{n_S \times m_S}$ ($n_S \leq n_T, m_S \leq m_T$). The extraction of these submatrices can be conducted through different methods: pass-by-neuron, by selecting rows and columns, or by direct extraction of the submatrix. Our comparative analysis in Section 4.3 indicates that maintaining the original structural integrity of the teacher model's parameters is most effective. Consequently, we compute the sensitivity scores for all submatrices with dimensions $\mathbb{R}^{n_S \times m_S}$. The submatrix $\boldsymbol{W}_{T_i,\text{extract}}$ is then selected as the one with the highest cumulative sensitivity score among these. Formally, this objective is expressed as:

$$\boldsymbol{W}_{T_i,\text{extract}} = \arg \max_{\boldsymbol{W}' \subseteq \boldsymbol{W}_{T_i}} \sum_{\theta_i \in \boldsymbol{W}'} S_i. \tag{5}$$

By aggregating $\boldsymbol{W}_{T_i,\text{extract}}$ across all layers, we derive the extracted parameters $\boldsymbol{\Theta}_{T_{\text{extract}}}$ from M_T.

3.3 Knowledge Injection

To keep the architecture and the number of parameters of the student model unchanged during the knowledge transfer process, we employ the LoRA approach to instantiate the Inject$(\cdot)$ function.

LoRA Module. LoRA (Hu et al., 2022), which stands for Low-Rank Adaptation, is a method designed to optimize parameter efficiency by freezing the pre-trained model weights and inserting trainable rank decomposition matrices into the deep neural network. The guiding intuition is that pre-trained language models possess low "intrinsic dimensions" (Aghajanyan et al., 2021). This means that even when projected to a smaller subspace, these models can still exhibit efficient learning. Consequently, it can be hypothesized that weight updates during adaptation also exhibit low "intrinsic ranks". For a given pre-trained weight matrix $\boldsymbol{W}_i \in \mathbb{R}^{n \times m}$, it can be updated as:

$$\boldsymbol{W}_i^* = \boldsymbol{W}_i + \Delta \boldsymbol{W} = \boldsymbol{W}_i + \boldsymbol{B}_i \boldsymbol{A}_i, \quad \boldsymbol{B}_i \in \mathbb{R}^{n \times r}, \quad \boldsymbol{A}_i \in \mathbb{R}^{r \times m}, \tag{6}$$

where r represents the low rank with $r \ll \min(n, m)$. The matrix $\boldsymbol{W}_i$ remains constant during this operation, implying that only $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ are updated in the training phase. To ensure that training commences from the original pre-trained weights, either $\boldsymbol{B}_i$ or $\boldsymbol{A}_i$ is initialized with zeros.

Knowledge Injection with LoRA. The main goal of this step is to integrate knowledge from the teacher model by incorporating extracted parameters into the student's LoRA module. Initially, SVD is adopted to factorize each matrix $\boldsymbol{W}_{T_i,\text{extract}}$ in $\boldsymbol{\Theta}_{T_{\text{extract}}}$ into three constituent matrices as:

$$\boldsymbol{W}_{T_i,\text{extract}} = \boldsymbol{U}_i \boldsymbol{\Sigma}_i \boldsymbol{V}_i^\top. \tag{7}$$

Here, $\boldsymbol{U}_i$ and $\boldsymbol{V}_i^\top$ are orthogonal matrices containing left and right singular vectors, respectively, while $\boldsymbol{\Sigma}_i$ is a diagonal matrix that hosts the singular values in descending order. To capture the first r ranks, we then formulate:

$$\boldsymbol{W}_{T_i,\text{extract},r} = \boldsymbol{U}_i[:, :r] \boldsymbol{\Sigma}_i[:r, :r] \boldsymbol{V}_i^\top[:r, :]. \tag{8}$$

The symbols $\boldsymbol{U}_i[:, :r]$ and $\boldsymbol{V}_i^\top[:r, :]$ represent the initial r columns of $\boldsymbol{U}_i$ and $\boldsymbol{V}_i^\top$, respectively, while $\boldsymbol{\Sigma}_i[:r, :r]$ captures the top r singular values. For the student model's corresponding matrix $\boldsymbol{W}_i$, we can adopt the training strategy from the LoRA paper:

$$\boldsymbol{W}_i^* = \boldsymbol{W}_i + \boldsymbol{B}_i \boldsymbol{A}_i, \tag{9}$$

where $\boldsymbol{B}_i$ is initialized as $\boldsymbol{U}_i[:, :r]\boldsymbol{\Sigma}_i[:r, :r]$, and $\boldsymbol{A}_i$ with $\boldsymbol{V}_i^\top[:r, :]$. This approach, however, alters the starting point from the pre-trained weights of the student model, potentially impacting downstream task performance, as discussed in Section 4.3. Consequently, we propose an alternative initialization strategy for the student model:

$$\boldsymbol{W}_i^* = \boldsymbol{W}_i - \boldsymbol{W}_{T_i,\text{extract},r} + \boldsymbol{B}_i\boldsymbol{A}_i, \tag{10}$$

During the training process, we maintain the matrices $\boldsymbol{W}_i$ and $\boldsymbol{W}_{T_i,\text{extract},r}$ as constants, with updates only being applied to the parameters in $\boldsymbol{B}_i\boldsymbol{A}_i$. Given that $\boldsymbol{W}_{T_i,\text{extract},r}$ and $\boldsymbol{B}_i\boldsymbol{A}_i$ are initially equivalent, this approach guarantees that training commences from the pre-trained weights. The inclusion of the LoRA module is designed to efficiently utilize the most salient features of the extracted knowledge from the teacher model.

4 Experiments

4.1 Experimental Setup

Facets of Evaluation. To validate the efficacy of our proposed framework, we conduct evaluations across four distinct benchmark categories:

(1) Reasoning: Reasoning stands as a foundational capability for models, particularly when tackling intricate tasks. We leverage the Grade School Math dataset (GSM) (Cobbe et al., 2021) to assess the reasoning proficiency of models. The evaluation format requires models, given a math problem, to produce the chain-of-thought process (Wei et al., 2022) and the final numerical answer.

(2) Professional Knowledge: For language models to effectively cater to users' informational needs, possessing a robust repository of professional knowledge is crucial. We measure this knowledge reservoir using the Massive Multitask Language Understanding dataset (MMLU) (Hendrycks et al., 2021). This dataset encompasses questions about 57 subjects, spanning a spectrum of difficulty levels from elementary to advanced professional tiers, all formatted as multiple-choice questions.

(3) Instruction-driven NLP Tasks: This set of tasks evaluates a model's capability in adhering to instructions. Typically, the language model receives both a task definition and an input text, and it must perform the specified classification or generation tasks as directed. Our chosen benchmark for this category is the Super Natural Instructions (Super NI) (Wang et al., 2022), a rich dataset comprising 1,616 varied NLP tasks alongside their expert-written instructions.

(4) Open-ended Conversation: This represents the primary interface through which models interact with users in real-world scenarios. To evaluate such instructability, we employ AlpacaFarm (Dubois et al., 2023), which contains 805 instructions including subsets from various evaluations like Self-Instruct (Wang et al., 2023c), Open Assistant (Köpf et al., 2023), Anthropic (Bai et al., 2022), Vicuna (Chiang et al., 2023), and Koala (Geng et al., 2023). GPT4-32K serves as the evaluator, determining the win rate of the testing model against the outputs generated by Davinci-003.

Throughout all evaluations, we adhere to established metrics and prompts, utilizing the evaluation scripts sourced from Wang et al. (2023b).

Implementation Details. For all our experiments, the larger-scale LLaMA model (Touvron et al., 2023a;b) serves as the teacher, and its smaller-scale counterpart acts as the student. For the fine-tuning of the student model, we draw a random subset of 1,000 instances from the respective training datasets of each benchmark. In the case of AlpacaFarm, due to the absence of a training set, we utilize LIMA data (Zhou et al., 2023) as a substitute, which is composed of 1,000 carefully curated open-ended conversations. For each experiment, 32 seed samples are randomly selected from the corresponding training sets. The student model is trained for 3 epochs with a batch size of 64 and a learning rate of 3e-4. Regarding LoRA, we set the rank as 16, and insert LoRA module into the embedding layer, FFN, and self-attention layer in the Transformer architecture (Vaswani et al., 2017). Notably, all results presented in this paper are mean values derived from three separate runs, with each run using a new random set of seed samples.

Table 1: Results for parametric knowledge transfer. "7B-LoRA + 13B Param." represents that we extract parameters from the 13B teacher model and transfer them to the 7B student model.

Models	GSM		MMLU		Super NI		AlpacaFarm	Average
	0-shot	8-shot	0-shot	5-shot	EM	R-L	Win Rate%	-
LLaMA-1								
Vanilla 7B	4.70	10.77	32.10	35.30	0.67	5.55	-	-
7B-LoRA	17.26	16.93	43.43	38.90	22.91	40.49	9.07	27.00
+ 13B Param.	**18.73**	**18.85**	44.03	39.77	24.51	42.37	9.28	28.22
+ 30B Param.	18.63	18.52	**45.20**	**40.60**	**25.01**	**43.08**	**9.40**	**28.63**
Vanilla 13B	4.93	17.44	43.50	46.80	2.18	7.78	-	-
13B-LoRA	26.18	23.78	50.43	50.03	27.34	45.53	13.91	33.89
+ 30B Param.	**27.85**	**27.70**	**51.30**	**51.03**	**27.51**	**46.09**	**17.27**	**35.54**
LLaMA-2								
Vanilla 7B	3.34	15.54	41.70	45.80	0.00	4.68	-	-
Vanilla 13B	6.52	27.82	52.10	55.20	0.00	4.84	-	-
7B-LoRA	23.38	21.05	47.77	**47.07**	24.93	41.25	20.50	32.28
+ 13B Param.	**25.30**	**26.31**	**49.37**	46.53	**26.16**	**42.98**	**24.64**	**34.47**

4.2 Experimental Results

Results for Parametric Knowledge Transfer. Our initial experiments focus on four distinct teacher-student model pairings: LLaMA-1 (13B $\Rightarrow$ 7B, 30B $\Rightarrow$ 7B, 30B $\Rightarrow$ 13B) and LLaMA-2 (13B $\Rightarrow$ 7B). The outcomes are systematically presented in Table 1. Remarkably, in contrast to the direct fine-tuning approach of LoRA, the student models augmented with parametric knowledge from their respective teacher models exhibit substantial improvements across all four benchmark categories. For instance, the LLaMA-1 30B $\Rightarrow$ 7B pairing yields an average performance boost of 6.04% (from 27.00 to 28.63). In a similar vein, the LLaMA-2 13B $\Rightarrow$ 7B configuration brings an enhancement of 6.78% (from 32.28 to 34.47).

Another observation emerges when examining the effects of scaling up the teacher model, specifically transitioning from 13B to 30B. The performance of the student model, LLaMA-1 7B, generally sees an improvement, despite a slight decrement in the GSM benchmark. Beyond the evident performance gains, the overhead introduced by parametric knowledge transfer remains minimal. The only extra commitment involves the teacher model executing inference on a set of 32 seed samples, without any direct participation in the training. Considering both performance and efficiency, parametric knowledge transfer stands out as a practical technique, even as disparities in parameter counts and architectural variances between teacher and student models expand.

Table 2: Transfer experiments with different task-specific extracted parameters. The leftmost column indicates the dataset on which the knowledge extraction is based. The teacher model and student model are LLaMA-2 13B and 7B, respectively.

Models	GSM		MMLU		Super NI		AlpacaFarm	Average
	0-shot	8-shot	0-shot	5-shot	EM	R-L	Win Rate%	-
Vanilla 7B	3.34	15.54	41.70	45.80	0.00	4.68	-	-
7B-LoRA	23.38	21.05	47.77	47.07	24.93	41.25	20.50	32.28
GSM	**25.30**	**26.31**	48.40	45.97	24.45	42.11	23.68	33.75
MMLU	24.11	25.47	**49.37**	46.53	25.55	42.55	24.01	33.94
Super NI	23.78	24.11	48.60	46.70	**26.16**	**42.98**	24.31	33.81
LIMA	24.08	25.60	49.03	**47.23**	25.63	42.83	**24.64**	**34.15**

Transfer Experiments with Task-specific Extracted Parameters. While our results indicate that transferring extracted knowledge from the teacher model positively influences student model performance, the nature of this improvement—whether it is rooted in generalized knowledge or task-specific expertise—warrants deeper exploration. To disentangle this, we conduct experiments

wherein extracted parameters, each tailored to a specific task, are integrated into the student model, which is subsequently fine-tuned across all datasets.

Table 2 offers insights into a prevalent trend: when parameters are extracted from a concrete task, the performance is most significantly amplified for that same task. This is particularly evident in the GSM benchmark. Models equipped with GSM-oriented extracted parameters notably exceed their counterparts—achieving at least a 1.2 increase in 0-shot accuracy—compared to models incorporated with parameters based on alternative datasets. This is likely due to the unique and intricate challenges associated with mathematical reasoning. Additionally, parameters sourced from the LIMA dataset demonstrate remarkable generalizability, presumably owing to their grounding in open-ended dialogues that cover a spectrum of domains and tasks. Overall, these observations highlight the capability of our sensitivity-driven techniques to efficiently target certain types of knowledge, rather than just extracting generalized knowledge.

4.3 Analysis: Key Factors for Parametric Knowledge Transfer

We further analyze the key factors for the process of parametric knowledge transfer as follows.

Initialization Strategies. Our analysis begins with a comparison of two initialization strategies: the approach described in the LoRA paper (Equation 9) and our proposed method (Equation 10). We employ the LLaMA-1 7B as the student model and explore 5 methods to initialize its LoRA module. These include Gaussian initialization for both $\boldsymbol{B}$ and $\boldsymbol{A}$ matrices, random extraction of sub-matrices from the 13B and 30B models, and sensitivity score-based extraction of sub-matrices from both the 13B and 30B models.

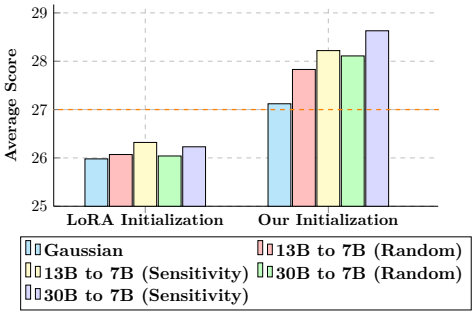


Figure 3: Comparison of different initialization strategies. The y-axis represents the average score over four datasets. "13B to 7B (Sensitivity)" refers to initializing the LoRA module in the 7B model with submatrices from the 13B model based on sensitivity score. The orange dotted line denotes the result of fine-tuning 7B-LoRA without knowledge transfer.

We present our findings in Figure 3. Initializing as per the LoRA paper—but without zeroing out $\boldsymbol{BA}$—leads to a noticeable drop in performance. Recognizing the imperative of leveraging the original model's weights as a starting point for fine-tuning the LoRA, our initialization strategy in this paper is rooted in Equation 10; hence, we keep both $\boldsymbol{W}$ and $\boldsymbol{W}_{\text{extract}}$ fixed and solely fine-tune $\boldsymbol{BA}$. Moreover, our sensitivity-based method consistently outperforms both Gaussian initialization and random parameter extraction from teacher models across varying scales.

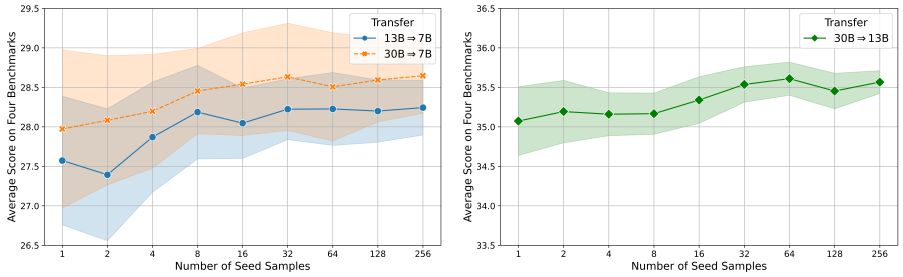
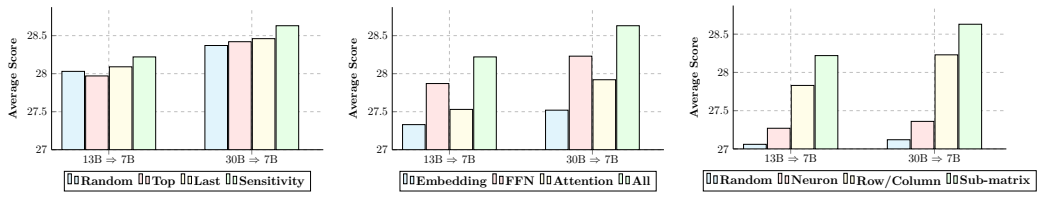


Figure 4: Analysis of how the quantity of seed samples affects student performance.

Number of Seed Samples. The quantity of seed samples plays a crucial role in determining both the reliability and efficiency of computing sensitivity scores from the teacher model. To delve deeper

into its impact, we study how varying numbers of seed samples influence the performance of the student model. As evidenced in Figure 4, an augmentation in seed samples consistently mitigates variance, whilst the enhancement in performance remains relatively slight. The results demonstrate a tendency to stabilize after the application of 32 seed samples, prompting us to establish this as a hyperparameter in this paper. A further insight is the marked reduction in variance as the student's scale is escalated (transitioning from 30B $\Rightarrow$ 7B to 30B $\Rightarrow$ 13B), or as the disparity between the teacher and student models is diminished (transitioning from 30B $\Rightarrow$ 7B to 13B $\Rightarrow$ 7B).



(a) Layer Selection (b) Origin of Parameters (c) Structure of Parameters

Figure 5: Analysis of various aspects of extracted parameters from teacher models. The y-axis begins with the result of direct fine-tuning students without knowledge injection.

Layer Selection Methods. Owing to the discrepancy in the number of layers between teacher and student models, the selection methodology for these layers potentially influences the final results. We evaluate four strategies: random layer selection, extracting the top or last layers, and a selection based on our sensitivity-centric technique. In the experiments, we consistently map teacher layers to student layers in their inherent sequential order. As Figure 5(a) illustrates, while the layer selection modestly affects student performance, our sensitivity-driven approach excels over the other strategies across both teacher-student model pairings.

Origin of Extracted Parameters. The complex architecture of the Transformer raises inquiries about the most effective module for knowledge transfer. Our explorations involve the Embedding layer, the Feed-Forward Network (FFN), and the Self-Attention layer of the teacher model. As depicted in Figure 5(b), the embedding layer experiences inferior transfer effectiveness, likely due to its lesser parameter quantity. In contrast, the FFN showcases advanced transfer capabilities, intimating that it houses a significant share of the teacher's knowledge. Optimal results are obtained when transferring knowledge from all available modules.

Structure of Extracted Parameters. The necessity to reduce the parameter matrix's size for knowledge transfer prompts questions regarding optimal population strategies for this matrix. We undertake a comparison across four methods: random single-weight selection from the teacher model, and parameter extraction based on the highest sensitivity at the single weight, row and column, and submatrix levels. Figure 5(c) shows that maintaining the teacher model's parameter structure significantly benefits student model performance. More precisely, transferring isolated single weights—either randomly or based on sensitivity—yields results comparable to those without knowledge transfer, highlighting the ineffectiveness of such transfers. Preserving the coherence of rows or columns provides a notable improvement, and the preservation of the submatrix structure further augments the performance gains derived from parametric knowledge transfer. This observation underpins our proposed knowledge extraction approach as outlined in Equation 5.

5 Conclusion

In this paper, we delve into the feasibility of transferring parametric knowledge between LLMs of varying scales, and present a new paradigm, exploring knowledge transfer from a distinct parametric perspective. Through our two-stage framework encompassing knowledge extraction and injection, we perform extensive experiments across four diverse benchmarks, affirming the inherent transferability of model parameters. Furthermore, by meticulously analyzing the key elements influencing parametric knowledge transfer, we aim to shed light on future research in this domain.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 7319–7328. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.568. URL https://doi.org/10.18653/v1/2021.acl-long.568.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=CQsmMYmlP5T.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022. doi: 10.48550/arXiv.2204.05862. URL https://doi.org/10.48550/arXiv.2204.05862.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 6491–6506. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.522. URL https://doi.org/10.18653/v1/2021.emnlp-main.522.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation, 2023. URL https://github.com/sahil280114/codealpaca.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2bert: Towards reusable pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2134–2148. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.151. URL https://doi.org/10.18653/v1/2022.acl-long.151.

Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05641.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. Blog post, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022. doi: 10.48550/arXiv.2204.02311. URL https://doi.org/10.48550/arXiv.2204.02311.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. *CoRR*, abs/2309.10668, 2023. doi: 10.48550/arXiv.2309.10668. URL https://doi.org/10.48550/arXiv.2309.10668.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *CoRR*, abs/2305.14387, 2023. doi: 10.48550/arXiv.2305.14387. URL https://doi.org/10.48550/arXiv.2305.14387.

Utku Evci, Bart van Merrienboer, Thomas Unterthiner, Fabian Pedregosa, and Max Vladymyrov. Gradmax: Growing neural networks using gradient information. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=qjN4h_wwUO.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. Blog post, April 2023. URL https://bair.berkeley.edu/blog/2023/04/03/koala/.

Andrea Gesmundo and Kaitlin Maile. Composable function-preserving expansions for transformer architectures. *CoRR*, abs/2308.06103, 2023. doi: 10.48550/arXiv.2308.06103. URL https://doi.org/10.48550/arXiv.2308.06103.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 129(6):1789–1819, 2021. doi: 10.1007/s11263-021-01453-z. URL https://doi.org/10.1007/s11263-021-01453-z.

Xiaotao Gu, Liyuan Liu, Hongkun Yu, Jing Li, Chen Chen, and Jiawei Han. On the transformer growth for progressive BERT training. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5174–5180. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.406. URL https://doi.org/10.18653/v1/2021.naacl-main.406.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 14409–14428. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.806. URL https://doi.org/10.18653/v1/2023.acl-long.806.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJYwwY9ll.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=6t0Kwf8-jrj.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Amir Globerson and Ricardo Silva (eds.), *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pp. 876–885. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/313.pdf.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438, 2020. doi: 10.1162/tacl_a_00324. URL https://doi.org/10.1162/tacl_a_00324.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 1345–1354. IEEE, 2019. doi: 10.1109/ICCV.2019.00143. URL https://doi.org/10.1109/ICCV.2019.00143.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 1317–1327. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1139. URL https://doi.org/10.18653/v1/d16-1139.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations - democratizing large language model alignment. *CoRR*, abs/2304.07327, 2023. doi: 10.48550/arXiv.2304.07327. URL https://doi.org/10.48550/arXiv.2304.07327.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7528–7538, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/94ef7214c4a90790186e255304f8fd1f-Abstract.html.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Xuying Meng, Siqi Fan, Peng Han, Jing Li, Li Du, Bowen Qin, Zheng Zhang, Aixin Sun, and Yequan Wang. FLM-101B: an open LLM and how to train it with $100k budget. *CoRR*, abs/2309.03852, 2023. doi: 10.48550/arXiv.2309.03852. URL https://doi.org/10.48550/arXiv.2309.03852.

Chen Liang, Haoming Jiang, Simiao Zuo, Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. No parameters left behind: Sensitivity guided adaptive learning rate for training large transformer models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=cuvga_CiVND.

Ekdeep Singh Lubana and Robert P. Dick. A gradient flow framework for analyzing network pruning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=rumv7QmLUue.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=MkbcAHIYgyS.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5191–5198. AAAI Press, 2020. doi: 10.1609/aaai.v34i04.5963. URL https://doi.org/10.1609/aaai.v34i04.5963.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning*

Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJGCiw5gl.

Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. *Advances in neural information processing systems*, 1, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/hash/07e1cd7dca89a1678042477183b7ac3f-Abstract.html.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/arXiv.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *CoRR*, abs/2305.12827, 2023. doi: 10.48550/arXiv.2305.12827. URL https://doi.org/10.48550/arXiv.2305.12827.

Dae Young Park, Moon-Hyun Cha, Changwook Jeong, Daesin Kim, and Bohyung Han. Learning student-friendly teacher networks for knowledge distillation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 13292–13303, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6e7d2da6d3953058db75714ac400b584-Abstract.html.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277, 2023. doi: 10.48550/arXiv.2304.03277. URL https://doi.org/10.48550/arXiv.2304.03277.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2227–2237. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1202. URL https://doi.org/10.18653/v1/n18-1202.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1250. URL https://doi.org/10.18653/v1/D19-1250.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 11557–11568. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01139. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pham_Meta_Pseudo_Labels_CVPR_2021_paper.html.

Yujia Qin, Yankai Lin, Jing Yi, Jiajie Zhang, Xu Han, Zhengyan Zhang, Yusheng Su, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Knowledge inheritance for pre-trained language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 3921–3937. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.288. URL https://doi.org/10.18653/v1/2022.naacl-main.288.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 5418–5426. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.437. URL https://doi.org/10.18653/v1/2020.emnlp-main.437.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

Sheng Shen, Pete Walsh, Kurt Keutzer, Jesse Dodge, Matthew E. Peters, and Iz Beltagy. Staged training for transformer language models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19893–19908. PMLR, 2022. URL https://proceedings.mlr.press/v162/shen22f.html.

George Stoica, Daniel Bolya, Jakob Bjorner, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. *CoRR*, abs/2305.03053, 2023. doi: 10.48550/arXiv.2305.03053. URL https://doi.org/10.48550/arXiv.2305.03053.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a. doi: 10.48550/arXiv.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b. doi: 10.48550/arXiv.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Peihao Wang, Rameswar Panda, Lucas Torroba Hennigen, Philip Greengard, Leonid Karlinsky, Rogério Feris, David Daniel Cox, Zhangyang Wang, and Yoon Kim. Learning to grow pretrained models for efficient transformer training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/pdf?id=cDYRS5iZ16f.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.340. URL https://doi.org/10.18653/v1/2022.emnlp-main.340.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. *CoRR*, abs/2306.04751, 2023b. doi: 10.48550/arXiv.2306.04751. URL https://doi.org/10.48550/arXiv.2306.04751.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics, 2023c. doi: 10.18653/v1/2023.acl-long.754. URL https://doi.org/10.18653/v1/2023.acl-long.754.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. Network morphism. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 564–572. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/wei16.html.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402, 2023. doi: 10.48550/arXiv.2304.14402. URL https://doi.org/10.48550/arXiv.2304.14402.

Canwen Xu, Daya Guo, Nan Duan, and Julian J. McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *CoRR*, abs/2304.01196, 2023. doi: 10.48550/arXiv.2304.01196. URL https://doi.org/10.48550/arXiv.2304.01196.

Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. In *International Conference on Learning Representations (ICLR)*, 2024.

Yiqun Yao, Zheng Zhang, Jing Li, and Yequan Wang. 2x faster language model pre-training via masked structural growth. *CoRR*, abs/2305.02869, 2023a. doi: 10.48550/arXiv.2305.02869. URL https://doi.org/10.48550/arXiv.2305.02869.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *CoRR*, abs/2305.13172, 2023b. doi: 10.48550/arXiv.2305.13172. URL https://doi.org/10.48550/arXiv.2305.13172.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operations. *CoRR*, abs/2306.14870, 2023. doi: 10.48550/arXiv.2306.14870. URL https://doi.org/10.48550/arXiv.2306.14870.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4320–4328. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00454. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Deep_Mutual_Learning_CVPR_2018_paper.html.

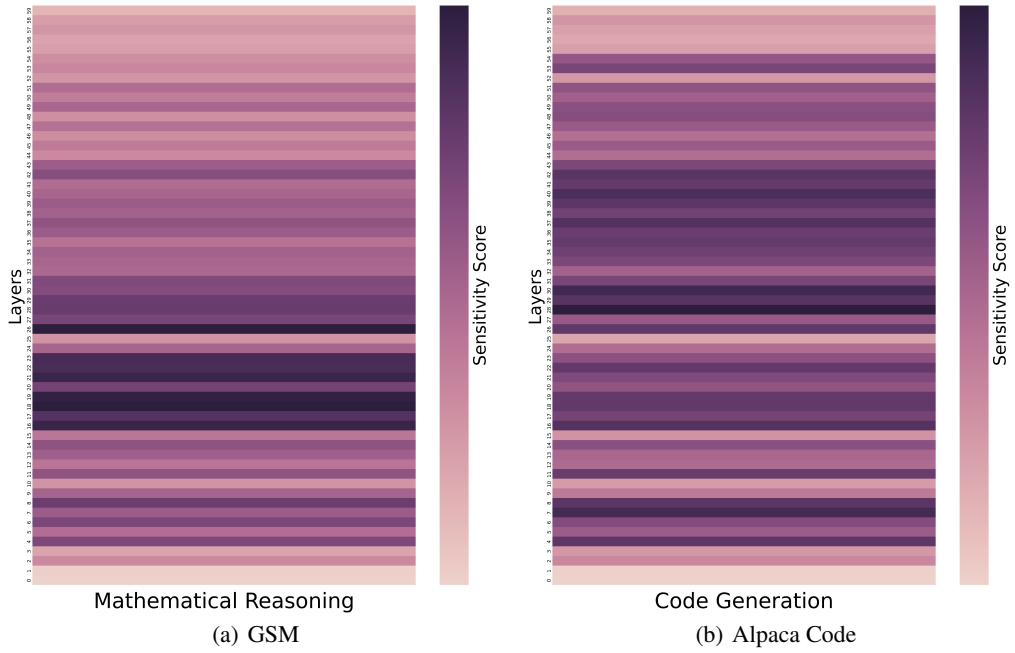
Ming Zhong, Yelong Shen, Shuohang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. *CoRR*, abs/2305.11206, 2023. doi: 10.48550/arXiv.2305.11206. URL https://doi.org/10.48550/arXiv.2305.11206.

Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. BERT learns to teach: Knowledge distillation with meta learning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 7037–7049. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.485. URL https://doi.org/10.18653/v1/2022.acl-long.485.

A Appendix

A.1 Visualization for Parametric Knowledge



(a) GSM (b) Alpaca Code

(c) Super NI (d) LIMA

Figure 6: Visualization of parametric knowledge across different layers for four distinct task categories. Darker shades represent higher sensitivity scores for each layer.

In our exploration, we also attempt to visualize the parametric knowledge intrinsic to different task categories. MMLU is omitted from the set of tasks, given its encompassing knowledge from multiple domains, and we introduce code generation (Chaudhary, 2023) as an additional task for analysis. LLaMA-1 30B serves as the teacher model, and we base our findings on 32 randomly selected seed samples, illustrating the sensitivity scores layer by layer. During the visualization process, we sub-

ject each parameter matrix to min-max normalization, ensuring that sensitivity scores fall within the [0, 1] range. The insights from Figure 6 reveal that the distribution of parametric knowledge across layers varies considerably among tasks. For instance, mathematical reasoning predominantly engages the bottom layer, instruction-driven NLP tasks concentrate on the bottom and middle layers, open-ended conversations are more centered around the middle and upper layers, while code generation appears to draw from all layers. This further emphasizes the efficacy of our sensitivity-based knowledge extraction method in pinpointing task-specific parametric knowledge, thereby aiding the subsequent transfer processes between diverse models.

A.2 Extended Experiment on LLaMA-2 70B to 7B

Table 3: Results for parametric knowledge transfer on LLaMA-2, with additional 70B to 7B transfer results included.

Models	GSM (0-shot)	MMLU (0-shot)	Super NI (R-L)	AlpacaFarm	Average
Vanilla 7B	3.34	41.70	4.68	-	-
Vanilla 13B	6.52	52.10	4.84	-	-
7B-LoRA	23.38	47.77	41.25	20.50	33.23
+ 13B Param.	25.30	49.37	42.98	**24.64**	**35.57**
+ 70B Param.	**26.16**	**49.60**	**43.65**	24.27	**35.92**

As shown in Table 3, we additionally exhibit the LLaMA-2 70B to 7B experiment for reference. Due to the differences in attention architectures between LLaMA-2 70B and 7B models (with the 70B employing grouped-query attention and the 7B using multi-head attention), we restrict the parametric knowledge transfer from 70B to 7B to the FFN and embedding layers, which account for 0.36% trainable parameters. In contrast, in the transfer from the 13B to 7B model, we also include the attention module, increasing the percentage of trainable parameters to 0.61%. Nevertheless, the transfer from the 70B to the 7B model demonstrates greater performance gains than transferring from 13B to 7B. This implies that our parametric knowledge transfer approach becomes increasingly effective as the teacher model scales up, even in the presence of architectural differences beyond the number of layers and dimensions.

A.3 Analysis of the Number of Parameters

Table 4: Analysis of the effect of the number of parameters on the performance of parametric knowledge transfer.

Transfer Module	LoRA r	Params.	13B to 7B	30B to 7B
Embedding	128	0.137%	27.33	27.52
Attention	16	0.248%	27.53	27.92
FFN	16	0.343%	27.87	28.23
All Modules	4	0.152%	27.93	28.37
All Modules	8	0.304%	28.17	28.48
All Modules	16	0.608%	28.22	**28.63**
All Modules	32	1.216%	28.19	28.54
All Modules	64	2.432%	**28.27**	28.60

Our analysis of the origin of parameters (see Figure 5(b)) involves a discussion of parameter amounts. For example, transferring a combination of the embedding layer, FFN, and attention modules, which constitute 0.608% of the model's parameters, yields the best results. In contrast, transferring a single module, like the FFN alone which accounts for 0.343% trainable parameters, leads to relatively poor performance.

To further investigate the effect of the number of parameters, we extend the experiments at LLaMA-1 13B to 7B and 30B to 7B by introducing comparisons with different LoRA r (intrinsic rank). The results of the average score on the four datasets are listed in Table 4. For the transfer involving

only the embedding layer, we set LoRA r at 128, ensuring that the number of trainable parameters remains comparable. We observe that increasing LoRA r beyond 16 does not significantly enhance the results when transferring all modules. Consequently, we maintain LoRA r at 16 for the experiments presented in this paper. Our analysis indicates that the origin of the parameters has a more pronounced impact on the effectiveness of parametric knowledge transfer than the number of parameters transferred.

A.4 Comparison with Distillation Methods

The parametric knowledge transfer paradigm in this paper is fundamentally distinct from traditional distillation methods, characterized by the following differences:

- **Purpose and Focus**: Rather than proposing better distillation methods, our study is focused on exploring the transferability of implicit knowledge embedded in static parameters. While prior research has explored the detectability and editability of parametric knowledge, its transferability remains less explored. Our experiments provide empirical evidence in the knowledge transfer scenario, where student models show improved performance after receiving task-specific knowledge from the teacher model, as shown in Tables 1 and 2.
- **Process and Efficiency**: Our approach differs from standard knowledge distillation, which typically requires fine-tuning or the direct involvement of the teacher model in student model training — a computationally intensive process. In contrast, our parametric knowledge transfer involves extracting task-specific parameters from the vanilla teacher model and integrating them into the student model. This method, requiring only 32 inferences from the teacher model, offers a significant efficiency advantage, especially in the context of LLMs.

Despite these differences, we provide the results of the distillation methods as a reference in the knowledge transfer scenario. Table 1 contains the results for LLaMA-1 13B to 7B and 30B to 7B, where KD refers to vanilla knowledge distillation (Hinton et al., 2015) and SeqKD is sequence-level knowledge distillation (Kim & Rush, 2016).

Table 5: Results for parametric knowledge transfer on LLaMA-1, additionally including results from knowledge distillation methods.

Models	GSM (0-shot)	MMLU (0-shot)	Super NI (R-L)	AlpacaFarm	Average
Vanilla 7B	4.70	32.10	5.55	-	-
Vanilla 13B	4.93	43.50	7.78	-	-
7B-LoRA	17.26	43.43	40.49	9.07	27.56
13B to 7B (KD)	17.69	43.57	42.08	9.32	28.17
13B to 7B (SeqKD)	17.86	43.33	41.91	9.36	28.12
13B to7B (Ours)	**18.73**	44.03	42.37	9.28	28.60
30B to 7B (KD)	17.81	44.10	41.96	9.48	28.34
30B to 7B (SeqKD)	17.99	43.97	42.40	**9.61**	28.49
30B to 7B (Ours)	18.63	**45.20**	**43.08**	9.40	**29.08**

All models are fine-tuned with LoRA, using identical training data and hyperparameters. In this paper, we attempt to explore the evidence that knowledge in static parameters can be transferred between different LLMs, and knowledge transfer is the scenario in which we find and provide empirical evidence. Our focus is not on proposing to find better methods in this scenario.

A.5 Trade-off Discussion between Performance and Running Cost

Considering that users may have varying computational resources in practical application scenarios, we discuss the trade-offs between performance and running costs as follows:

Experimental details:

- We conduct comparisons for two knowledge transfers: LLaMA-1 13B to 7B and 30B to 7B.

Table 6: Experimental results for the trade-off discussion between the performance and the running cost.

Transfer	Structure of Ext. Para.	Methods for Ext. Para.	Score	Inference Time	Memory
13B to 7B	Submatrix	Random	27.83	39 min	205 G
13B to 7B	Submatrix	Sensitivity	28.22	39 min	205 G
13B to 7B	Single Weights	Random	27.06	39 min	205 G
13B to 7B	Single Weights	Sensitivity	27.27	39 min	205 G
30B to 7B	Submatrix	Random	28.11	82 min	510 G
30B to 7B	Submatrix	Sensitivity	28.63	82 min	510 G
30B to 7B	Single Weights	Random	27.12	82 min	510 G
30B to 7B	Single Weights	Sensitivity	27.36	82 min	510 G

- For the structure of extracted parameters, "submatrix" extraction involves directly taking sub-matrices from the teacher model's larger matrix that aligns with the student model's dimensions. In contrast, "single weights" extraction means taking individual weights from the teacher model's larger matrix and arranging them into smaller matrices, maintaining their original order, to initialize the student model's LoRA module.
- We compare random selection with our sensitivity score-based method for choosing parameters.
- The "Score" column represents the average performance across four benchmarks.
- The running cost is compared on a CPU for 32 seed examples from GSM dataset, using teacher models of 13B and 30B. We perform backpropagation for each inference and based our experiments on fp32. Given that users may have limited GPU memory in real-world applications, we conduct these experiments on CPUs.

Key observations:

- The sensitivity score-based method we propose consistently outperforms random extraction.
- Extracting parameters via submatrices is significantly more effective than by single weights. This aligns with our discussion in Section 4.3 about the importance of maintaining the structural integrity of parameters for successful parametric knowledge transfer.
- Scaling up the teacher model from 13B to 30B consistently enhances performance but comes with about 2.1xthe runtime and 2.5x the memory usage.
- We can observe that the performance of transferring from 30B to 7B (Submatrix + Random) is comparable to (slightly lower than) 13B to 7B (Submatrix + Sensitivity). Thus, for applications with resource constraints, opting to randomly extract sub-matrices directly from a larger teacher model presents a viable alternative, considering it reduces inference time.