TITOK: TRANSFER TOKEN-LEVEL KNOWLEDGE VIA CONTRASTIVE EXCESS TO TRANSPLANT LORA

Chanjoo Jung[1], **Jaehyung Kim**[1]
[1]Yonsei University
chanjoo0427@yonsei.ac.kr, jaehyungk@yonsei.ac.kr

ABSTRACT

Large Language Models (LLMs) are widely applied in real world scenarios, but fine-tuning them comes with significant computational and storage costs. Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA mitigate these costs, but the adapted parameters are dependent on the base model and cannot be transferred across different backbones. One way to address this issue is through knowledge distillation, but its effectiveness inherently depends on training data. Recent work such as TransLoRA avoids this by generating synthetic data, but this adds complexity because it requires training an additional discriminator model. In this paper, we propose **TITOK**[1], a new framework that enables effective LoRA **T**ransplantat**i**on through **Tok**en-level knowledge transfer. Specifically, TITOK captures task-relevant information through a contrastive excess between a source model with and without LoRA. This excess highlights informative tokens and enables selective filtering of synthetic data, all without additional models or overhead. Through experiments on three benchmarks across multiple transfer settings, our experiments show that the proposed method is consistently effective, achieving average performance gains of + 4–8% compared to baselines overall.

1 INTRODUCTION

Large Language Models (LLMs) (Brown et al., 2020; Vaswani et al., 2023), have made significant progress in many real-world applications, including chatbots (OpenAI et al., 2024), search engines (Xiong et al., 2024), and coding assistants (Rozière et al., 2024). While fine-tuning LLMs has been demonstrated to be a promising way to improve performance on downstream tasks, it incurs substantial computational and storage costs. Parameter-Efficient Fine-Tuning (PEFT) (Houlsby et al., 2019) methods such as LoRA (Hu et al., 2021) alleviate this burden by updating only a small subset of parameters while keeping the base model frozen. However, PEFT's adapted parameters are dependent to the base model and cannot be transferred across different models. This limitation is increasingly critical given the rapid release of new LLMs and the growing diversity of available models.

One considerable approach to mitigate this limitation is through Knowledge distillation (KD) (Hinton et al., 2015; Azimi et al., 2024), which transfers the knowledge embedded in a source model's PEFT adapters to a target model with new PEFT adapters by aligning the target's output distributions with those of the source. However, KD is inherently data-dependent, typically requiring access to training data from target downstream tasks (Nayak et al., 2019; Liu et al., 2024), which is often unavailable or costly to obtain. To address this limitation, TransLoRA (Wang et al., 2024) has recently proposed using synthetic data by leveraging the data synthesis capabilities of recent LLMs (Wang et al., 2023; Kim et al., 2025). This approach enables the target model to acquire domain knowledge without direct access to the original dataset. Nevertheless, TransLoRA requires training an additional discriminator model to filter low-quality synthetic data, which inevitably introduces extra complexity and computational overhead. Furthermore, it primarily emphasizes the role of synthetic data, paying less attention to *how the knowledge transfer process itself should be designed*.

Contribution. In this paper, we propose a new framework that enables effective LoRA **T**ransplantat**i**on through **Tok**en-level knowledge transfer (**TITOK**). Our high-level idea is to se-

[1]Code will be released upon acceptance: https://github.com/NaughtyMaltiz16/TiTok

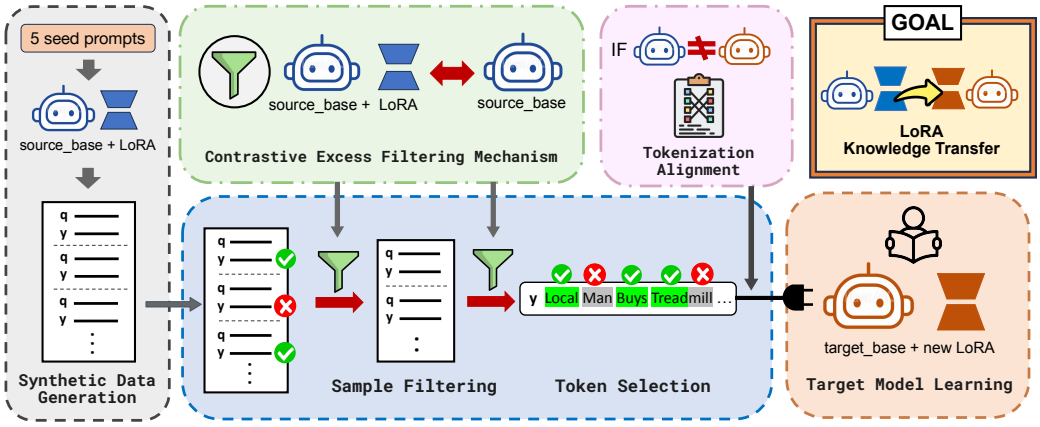


Figure 1: Overview of TITOK: Transplantation through Token-level knowledge transfer. Starting from a small set of seed prompts, the source expert model (source base model + LoRA) generates synthetic data. A contrastive excess filtering mechanism then compares the expert against its base backbone to compute token-level excess scores. Using these scores, TITOK first performs sample filtering and subsequently token selection, retaining only the most informative samples and tokens. When tokenizers differ, masks are aligned prior to training. The resulting filtered data is finally used to train a new LoRA on the target backbone, enabling efficient knowledge transfer.

lectively convey task-relevant information from the source model's LoRA by using token-level signals to guide the transfer process, rather than relying on the entire token sequence. We specifically capture this information through a concept we introduce as *contrastive excess*, obtained by comparing predictions from a source model with LoRA and the same model without LoRA. Intuitively, this contrastive excess highlights tokens that contain important task knowledge. This excess signal is further utilized to filter the synthetic data we generate for training, thereby enabling selective learning on samples that contain richer information. Unlike TransLoRA, which requires training and storing an additional discriminator model for filtering, TITOK requires neither extra models nor additional training overhead. Moreover, we design an effective mechanism to resolve tokenizer mismatches between source and target models, which enhances robustness and applicability of our framework.

We demonstrate the effectiveness of TITOK by conducting extensive experiments on three widely used benchmarks, which cover both reasoning (Big Bench Hard (Suzgun et al., 2022) and MMLU (Hendrycks et al., 2021)) and personalization tasks (LaMP (Salemi et al., 2024)). In particular, TITOK improves the performance by +7.96% over the vanilla target model, +6.0% over KD, and +4.4% over TransLoRA, when averaged across all tasks and transfer settings. We also explored a variety of transfer settings, including transfers within the same model, across different model families and sizes, and even across different model versions. In every case, our approach achieves consistent improvements. Interestingly, TITOK remains effective even when applied to external data originating from tasks different from the target task, highlighting both robustness and general applicability. Overall, these empirical results highlight TITOK as a methodologically simple yet powerful paradigm for efficiently transferring LoRA knowledge across models in diverse scenarios.

2 Related Works

Transferring PEFT adapters. Parameter-Efficient Fine-Tuning (PEFT) (Houlsby et al., 2019; Li & Liang, 2021) has emerged as both a practical and popular alternative to full model fine-tuning. By requiring updates to only a small portion of parameters, it enables efficient adaptation, with LoRA (Low-Rank Adaptation) (Hu et al., 2021; Dettmers et al., 2023) standing out as one of the most widely adopted methods. However, a fundamental limitation is that LoRA adapters are tied to the frozen backbone they were trained on, making them difficult to transfer to other base models. To address this issue, recent studies such as TransLoRA (Wang et al., 2024) have attempted to

transplant the knowledge of LoRA adapters across models by generating synthetic data (Wang et al., 2023). While effective to some extent, this approach requires an additional discriminator model to filter high-quality synthetic data, resulting in a relatively heavy pipeline. In parallel, Knowledge Distillation (KD) (Hinton et al., 2015; Azimi et al., 2024) has been widely explored as another means of knowledge transfer; however, traditional KD typically operates within a teacher–student framework at the logit or sequence level and requires access to training data close to the original, in order to use the teacher's distribution for supervising the student. In contrast, our approach focuses on **token-level selective transfer**, offering a more fine-grained yet lightweight alternative that enables efficient transplantation of LoRA adapters across diverse models for deployment.

Data synthesis with LLMs. Synthetic data generation with LLMs has attracted increasing attention as a means to reduce reliance on costly or inaccessible datasets (Wang et al., 2023; Kim et al., 2025). Prior lines of research have leveraged synthetic data for purposes such as privacy preservation (Bu et al., 2025), data augmentation (Kumar et al., 2021), and domain adaptation (Li et al., 2023). In our framework, synthetic data serves as a core component, enabling the transfer of LoRA adapters without access to the original training corpus, while simultaneously mitigating privacy concerns and reducing the degree of dependency on external datasets. Moreover, since both queries and labels are generated directly by the source expert model itself, synthetic data ultimately provides a self-sufficient mechanism that fits to our objective of lightweight and effective knowledge transfer.

Selective token training. Recent studies (Lin et al., 2025; Gu et al., 2020) demonstrate that not all tokens contribute equally to model training, motivating research on selective training strategies. While such approaches have primarily been applied to accelerate optimization or reduce redundancy (Yeongbin et al., 2024; Bal et al., 2025), our work innovatively extends this concept to the setting of knowledge transfer. Specifically, our concept of excess scores (Eq. 2) is derived from this idea, where the contrast between the source backbone and its LoRA adapter yields token-level judgments. This enables TITOK to transplant LoRA knowledge in a more focused and fine-grained manner, highlighting the broader applicability of selective training beyond its original scope.

3 TITOK: TRANSPLANTING LORA THROUGH TOKEN-LEVEL KNOWLEDGE

In this section, we introduce TITOK, a framework for LoRA **T**ransplantat**i**on through **Tok**en-level knowledge transfer (Fig. 1). The core idea of TITOK is to transfer the knowledge from a source model's LoRA adapter into a target model's LoRA adapter by training specifically on the informative tokens within synthetic data. Specifically, the framework consists of three components: (1) *Synthetic data generation* (Sec. 3.1), where the source expert model produces query–label pairs for target task; (2) *Excess score computation* (Sec. 3.2), which calculates token-level importance using source model; and (3) *Target model training with filtering* (Sec. 3.3), which trains target model with newly initialized LoRA adapter on top-ranked samples and tokens. In addition, we propose *Excess score alignment* (Sec. 3.4), an algortihm designed to apply TITOK even when tokenizers of the source and target models differ. The overall algorithm of TITOK is presented in Alg. 1.

3.1 SYNTHETIC DATA GENERATION VIA LLM PROMPTING WITH FEW-SHOT DATA

Let $\mathcal{M}_s$ denote the source backbone LLM and $\mathcal{A}_s$ its LoRA adapter on target downstream task, which forms the source expert model $\mathcal{M}_s + \mathcal{A}_s$. The target model, whose LoRA adapter $\mathcal{A}_t$ will be trained, is denoted by $\mathcal{M}_t$. Then, TITOK first constructs a synthetic dataset $\mathcal{D}_\text{s}$, similar to the idea of TransLoRA (Wang et al., 2024). This usage of synthetic data allows us to avoid keeping the whole original dataset for the downstream task, and simultaneously let $\mathcal{A}_t$ learn the knowledge encoded in the source adapter $\mathcal{A}_s$. Unlike TransLoRA's approach of using the untuned target model $\mathcal{M}_t$ to generate synthetic data, we use the source expert model $\mathcal{M}_s + \mathcal{A}_s$ to synthesize data (see empirical comparison in Fig. 2). Concretely, $\mathcal{M}_s + \mathcal{A}_s$ synthesizes both the *query* and the *label* within a prompting-based data synthesis framework (Wang et al., 2023) (see details in Appendix H); given a few-shot data of downstream task, it first generates a query $\mathbf{q}$, and then produces the corresponding label $\mathbf{y}$ conditioned on $\mathbf{q}$. To encourage diversity, we apply ROUGE-L filtering together with deduplication to all tasks, except for exceptional cases where such filtering is infeasible (more details are in Appendix G). Consequently, the resulting synthetic dataset consists of query–label pairs:

$$\mathcal{D}_\text{s} = \{(\mathbf{q}_j, \mathbf{y}_j)\}_{j=1}^{N}. \tag{1}$$

In this way, $\mathcal{M}_t$ would be trained on the synthetic data $\mathcal{D}_s$ generated by the source expert model $\mathcal{M}_s + \mathcal{A}_s$, thereby enabling knowledge transfer without relying on the entire original dataset.

3.2 CONTRASTIVE EXCESS SCORE FROM SOURCE MODEL WITH LORA ADAPTERS

As described in Sec. 3.1, TITOK relies on synthetic data, but synthetic data is often prone to imperfections (Chen et al., 2024); thus, sophisticated filtering is essential to retain high-quality informative samples. TransLoRA (Wang et al., 2024) tackles this challenge with a separate discriminator to filter useful queries, but this introduces the extra burden of training and maintaining an extra model. In contrast, we propose a lightweight alternative that only utilizes the already trained source model. Specifically, we use source model and its LoRA adapter to perform two complementary roles: (1) the *amateur* role ($\mathcal{M}_s$) and (2) the *expert* role ($\mathcal{M}_s + \mathcal{A}_s$). Then, the difference between the two roles provides an *implicit supervision signal* where the task information is encoded.

Formally, let $\mathbf{y} = [y_1, ..., y_L]$ denote the synthesized response, where y_i is a token of $\mathbf{y}$ corresponding to the synthesized query $\mathbf{q}$. Then, we define the *excess score* as:

$$S(y_i) = L_e(y_i) - L_a(y_i), \tag{2}$$

where the amateur and expert losses on token y_i are defined as:

$$L_a(y_i) = \log P_{\mathcal{M}s}(y_i \mid \mathbf{q}, \mathbf{y}{<}\, i), \quad L_e(y_i) = \log P_{\mathcal{M}_s+\mathcal{A}s}(y_i \mid \mathbf{q}, \mathbf{y}{<}\, i). \tag{3}$$

The excess score $S(y_i)$ quantifies the knowledge discrepancy incurred by equipment of LoRA, thereby identifying the tokens where the adapter provides a decisive contribution. Intuitively, if the backbone model is uncertain about predicting a token but the LoRA-enhanced model assigns it with high confidence, that token will thus obtain a large excess score. This implies that tokens with higher $S(y_i)$ correspond to positions where the LoRA adapter injects task-specific knowledge that the backbone could not capture on its own. In this way, the excess score functions as a *fine-grained attribution signal*, derived entirely from the internal behavior of the model itself, and guides training toward the specific regions of data that are most enriched with the adapter's knowledge.

3.3 TARGET MODEL TRAINING WITH SAMPLE FILTERING AND TOKEN SELECTION

After the computation of excess scores $S(y_i)$, the newly initialized LoRA adapter $\mathcal{A}_t$ for target model $\mathcal{M}_t$ is trained using the synthetic samples $(\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_\text{f}$ with two-level filtering schemes.

First stage: Sample filtering. We begin by filtering the synthetic dataset $\mathcal{D}_s$ (Eq. 1) at the sample level to remove less informative examples. For each synthetic sample, we compute the mean of the excess scores $S(y_i)$ across the tokens in $\mathbf{y}$ and retain only M samples with the highest values.

$$\bar{S}_j \;=\; \frac{1}{|\mathbf{y}_j|} \sum_{y_i \in \mathbf{y}_j} S(y_i). \tag{4}$$

Let $\mathcal{D}_\text{f}$ be the set of the M samples in $\mathcal{D}_\text{s}$ with the largest $\bar{S}_j$:

$$\mathcal{D}_\text{f} \;=\; \text{Top}M\big\{\; (\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_\text{s} \;:\; \bar{S}_j \;\big\}. \tag{5}$$

Through this step, the synthetic data undergoes a filtering process, ensuring that subsequent training is concentrated specifically on the remaining examples in $\mathcal{D}_\text{f}$ with richer knowledge signals.

Second stage: Token selection. Next, we consider token selection; that is, $\mathcal{A}_t$ does not learn from all tokens within the retained samples. Instead, it focuses only on those prioritized by the excess scores $S(y_i)$, which are identified as most important for knowledge transfer. To achieve this, we select the top $k\%$ of tokens ranked by their excess scores using the indicator $I_{k\%}(y_i)$:

$$I_{k\%}(y_i) = \begin{cases} 1, & \text{if } \text{rank}_{\mathbf{y}_j}\big(S(y_i)\big) \le \lfloor k\% \cdot |\mathbf{y}_j| \rfloor, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $|\mathbf{y}_j|$ denotes the number of tokens in $\mathbf{y}_j$, and $\text{rank}_{\mathbf{y}_j}(S(y_i))$ indicates the rank of $S(y_i)$ among the tokens of that response. Based on this selection, the training objective for $\mathcal{A}_t$ is defined as

$$\mathcal{L}_{\texttt{TiTok}} = \sum_{(\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_\text{f}} \; \sum_{y_i \in \mathbf{y}_j} I_{k\%}(y_i) \cdot L_t(y_i), \tag{7}$$

where $L_t(y_i)$ is the negative log-likelihood loss assigned by $\mathcal{M}_t + \mathcal{A}_t$ on token y_i (only $\mathcal{A}_t$ is learnable). By training only on these filtered tokens (Eq. 7), TITOK enables $\mathcal{A}_t$ to efficiently acquire the source LoRA's knowledge without access to the original training data or any external models.

Algorithm 1: TiTok: Transplanting LoRA through Token-Level Knowledge

Input: source expert $\mathcal{M}_s + \mathcal{A}_s$, target $\mathcal{M}_t$, parameters $N, M, k\%$
Output: trained target LoRA $\mathcal{A}_t$

1. Construct synthetic dataset $\mathcal{D}_\text{s} = \{(\mathbf{q}_j, \mathbf{y}_j)\}_{j=1}^{N}$ with $\mathcal{M}_s + \mathcal{A}_s$
2. **for** $(\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_s$ **do**
   Compute token excess scores $S(y_i) = L_e(y_i) - L_a(y_i)$
   Calculate mean score $\bar{S}_j = \frac{1}{|\mathbf{y}_j|} \sum_{y_i \in \mathbf{y}_j} S(y_i)$
3. Select top-M samples by $\bar{S}_j$ to form $\mathcal{D}_\text{f}$
4. **for** $(\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_f$ **do**
   Rank tokens by $S(y_i)$ and keep top-$k\%$, represented by mask $I_{k\%}(y_i)$
5. **if** *tokenizer*$(\mathcal{M}_s) \neq$ *tokenizer*$(\mathcal{M}_t)$ **then**
   Align masks $I^{(s)}_{k\%}(y_i) \rightarrow I^{(t)}_{k\%}(y_i)$
6. Train $\mathcal{A}_t$ on $\mathcal{M}_t$ with masked loss $\mathcal{L}_{\texttt{TiTok}} = \sum_{(\mathbf{q}_j, \mathbf{y}_j) \in \mathcal{D}_\text{f}} \sum_{y_i \in \mathbf{y}_j} I_{k\%}(y_i) \cdot L_t(y_i)$

return $\mathcal{A}_t$

3.4 Excess score alignment across different tokenizers

In cases of transfer between models with different tokenizers, a direct mapping of token-level signals is not possible, given that the source and target models may segment text differently. To address this, we introduce a simple yet robust tokenizer alignment algorithm that propagates token masks (Eq. 6) from the source token sequence $\mathbf{y}^{(s)}$ to the target token sequence $\mathbf{y}^{(t)}$. The algorithm first aligns token sequences using dual pointers that incrementally decode and match text spans. Masks are then propagated using the following four rules: (1) direct copy for one-to-one mappings, (2) replication for one-to-many, (3) averaging for many-to-one, and (4) averaging with replication for many-to-many. Finally, a top-$k\%$ selection step retains the most confident target tokens. This process ensures consistent supervision across tokenizers, enabling reliable transfer even when models tokenize text differently. The conceptual illustration of this procedure is presented in Fig.4.

4 Experiment

In this section, we present our experimental results to answer the following research questions:

- **RQ1:** Can TiTok efficiently transfer knowledge of LoRA in various scenarios? (Table 1)
- **RQ2:** What is the contribution of each component in TiTok? (Table 2)
- **RQ3:** How sensitive is token-level selective transfer to the selection ratio? (Figure 3)
- **RQ4:** How does the choice of model to synthesize query affect the performance? (Figure 2)
- **RQ5:** Can TiTok transfer knowledge using data from a different or unrelated domain? (Table 3)

4.1 Experimental setups

Models. We mainly present our knowledge transfer experiments using models from the Mistral and Llama families. Specifically, we designed the following LoRA transfer (*source* $\rightarrow$ *target*) setups. (1) Mistral-7B-Inst-v0.3[2] $\rightarrow$ Mistral-7B-Inst-v0.3: the basic transfer setup, (2) Mistral-7B-Inst-v0.3 $\rightarrow$ Llama-3.1-8B-Inst[3]: the different-family model transfer setup, (3) Llama-3.2-3B-Inst[4] $\rightarrow$ Llama-3.1-8B-Inst: the different-size model transfer setup, and (4) Llama-2-7b-chat-hf[5] $\rightarrow$ Llama-3.1-8B-Inst: the different-version model transfer setup. These setups are intended to test whether a smaller model can effectively transfer knowledge to a larger one, and to explore whether a relatively weaker model can still influence a newer, stronger model. These various setups realistically reflect how LLMs develop today, where various models are released and newer, improved versions keep

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[4] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[5] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

emerging. Model names are abbreviated for clarity and are used consistently in all tables and figures: 1) "Mistral 7B" = Mistral-7B-Instruct-v0.3, 2) "Llama3 8B" = Llama-3.1-8B-Instruct, 3) "Llama3 3B" = Llama-3.2-3B-Instruct, 4) "Llama2 7B" = Llama-2-7b-chat-hf.

Baselines. To demonstrate the effectiveness of TITOK, we compare it against three baselines: (i) *Vanilla*, the target base model without any fine-tuning; (ii) *KD(+MinED)*, knowledge distillation (KD) from source expert model (Hinton et al., 2015). When the source and target models use different tokenizers, the original KD is not applicable. In this case, we use Minimum-Edit-Distance (MinED) tokenizer alignment (Wan et al., 2024), which aligns both token sequences and vocabulary distributions via dynamic-programming sequence alignment for near matches (e.g., "gets" $\leftrightarrow$ "get") and by mapping probability mass to nearest edit-distance neighbors (e.g., "immediately" $\leftrightarrow$ "immediate"). We use the synthesized data by TransLoRA as the training data for KD and MinED ; and (iii) *TransLoRA* (Wang et al., 2024), a prior method in which the vanilla target model synthesizes the queries, while the source model with its LoRA adapter generates the corresponding synthetic labels. A discriminator is then employed to filter the synthetic data for training the target LoRA adapter.

Datasets. Following prior work in TransLoRA, we first conduct experiments on two representative benchmarks: (1) *Big-Bench Hard (BBH)* (Suzgun et al., 2022) and (2) *Massive Multitask Language Understanding (MMLU)* (Hendrycks et al., 2021). BBH consists of 27 challenging reasoning tasks structured as multiple choice or short answer questions, designed to test compositional generalization and advanced problem solving abilities of a Language Model. Meanwhile, MMLU covers 57 tasks across diverse academic subjects, presented in multiple choice format to evaluate broad knowledge and reasoning skills of a model. Since both benchmarks only provide test sets, we split the data into 90% for training the source expert model $\mathcal{M}_s + \mathcal{A}_s$ and the remaining 10% for evaluation.

To extend our approach to personalization and text generation, we additionally conduct experiments on LaMP benchmark (Salemi et al., 2024), focusing exclusively on its generation tasks. In particular, we experiment with *(3) News Headline Generation (LaMP 4)* and *(4) Scholarly Title Generation (LaMP 5)*, as they are the only text generation tasks that are both accessible and reliably evaluable. The remaining LaMP tasks are excluded, given that LaMP 1-3 are discriminative, and LaMP 6-7 lack gold labels. For LaMP tasks, the source expert model $\mathcal{M}_s + \mathcal{A}_s$ is trained on data from the 30 users with the longest activity histories. From each user, we use 200 data points for training and 50 data points for validation, a design choice intended to conduct a more rigorous and robust evaluation. In total, each LaMP task contains 6,000 training examples and 1,500 evaluation examples.

To assess performance on the BBH and MMLU, we measure the average accuracy using the LM-Eval Harness (Gao et al., 2024). Following the setup used in TransLoRA (Wang et al., 2024), all tasks are conducted in a zero-shot setting. Meanwhile, for the LaMP tasks, we adopt ROUGE-1 and ROUGE-L scores as evaluation metrics, following the benchmark's primary evaluation metrics.

Implementation details. For training of both source and target models, we use a learning rate of 5×10^{-5}, train for 2 epochs with a batch size of 4, and apply LoRA with rank $r = 8$, scaling factor $\alpha = 8$, and dropout 0.05. Optimization is performed using AdamW with weight decay 1×10^{-2}, together with a linear learning rate schedule and a warmup ratio of 0.1. For synthetic data generation, we provide five samples from the original training data as few-shot exemplars, and apply top-p sampling (Holtzman et al., 2020) to generate both queries and labels, with sampling hyperparameters tuned individually for each task. To further encourage diversity, we apply ROUGE-L filtering with a threshold of 0.7 and deduplication to remove redundant queries, following Wang et al. (2023). For tasks where ROUGE-based filtering is infeasible, we apply only deduplication (see Appendix G). For the initial synthetic pool, we generate $2M$ synthetic samples and, after filtering with contrastive excess scores, retain the top M, where M equals the source training set size (see Appendix F). For token selection, the selection ratio $k\%$ is fixed at 70% across all tasks and transfer settings, except for the Llama3 3B $\rightarrow$ Llama3 8B transfer setting, where we find that $k\% = 30\%$ consistently yields the best performance. For inference, we adopt greedy decoding to ensure deterministic evaluation.

4.2 MAIN RESULTS

Table 1 summarizes the experimental results on BBH, MMLU, and LaMP (News Headline and Scholarly Title generation) across four transfer settings. First, we observe that transfer within

Table 1: **Main results.** Experiments on BBH, MMLU, News Headline and Scholarly Title Generation tasks under four transfer settings. BBH and MMLU are reasoning tasks and are evaluated using LM-Eval Harness, while News Headline and Scholarly Title Generation represent personalization tasks and are evaluated with ROUGE-1/L. All evaluations are zero-shot. Best scores are in **bold**.

Transfer	Method	BBH	MMLU	News Headline		Scholarly Title	
		Acc.	Acc.	ROUGE-1	ROUGE-L	ROUGE-1	ROUGE-L
Mistral 7B → Mistral 7B	Vanilla	0.397	0.557	0.117	0.101	0.381	0.311
	KD	0.426	0.556	0.121	0.107	0.392	0.320
	TransLoRA	0.424	0.533	0.155	0.136	0.447	0.382
	TiTok (ours)	**0.432**	**0.563**	**0.160**	**0.142**	**0.473**	**0.414**
Mistral 7B → Llama3 8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD+MinED	0.477	0.484	0.127	0.112	0.455	0.389
	TransLoRA	0.471	0.472	0.122	0.108	0.461	0.397
	TiTok (ours)	**0.482**	**0.488**	**0.140**	**0.124**	**0.464**	**0.403**
Llama3 3B → Llama3 8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD	0.470	**0.475**	0.126	0.111	0.449	0.383
	TransLoRA	0.460	0.466	0.121	0.107	0.455	0.387
	TiTok (ours)	**0.509**	**0.475**	**0.127**	**0.113**	**0.457**	**0.392**
Llama2 7B → Llama3 8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD+MinED	0.473	0.478	0.125	0.111	0.450	0.384
	TransLoRA	0.472	0.468	0.123	0.109	0.453	0.388
	TiTok (ours)	**0.510**	**0.479**	**0.140**	**0.122**	**0.461**	**0.404**

the same model family is highly effective; when transplanting the LoRA adapter trained on the Mistral 7B into a fresh instance of the same model, TITOK consistently surpasses all baselines. In particular, TITOK improves the vanilla model by 24.08% on average across all tasks, and further outperforms the KD and TransLoRA baselines by 19.63% and 4.91%, respectively. Taken together, the findings show that, as a first step, transfer within the same family is reliably successful.

Beyond same family transfer, we also find that TITOK is highly effective in cross-model transfer settings. For example, when transferring from Mistral 7B to Llama 8B, TITOK delivers an average improvement across all tasks of 7.11% over the vanilla model, while outperforming KD by 4.73% and TransLoRA by 6.24%. This highlights that TITOK is not confined to intra-family knowledge transfer, but can successfully bridge across architectures. In the case of Llama3 3B to Llama3 8B, TITOK yields average gains of 3.45% against vanilla, 2.49% against KD, and 4.14% against TransLoRA. These results suggest that TITOK scales effectively with model size, making it useful when moving from lightweight to larger models without losing efficiency. Finally, when transferring from Llama2 7B to Llama3 8B, TITOK performs average advantages of 7.43% over vanilla, 6.28% over KD, and 7.22% over TransLoRA. This indicates that TITOK adapts robustly even across different model versions, implying practical relevance when models are upgraded in real-world pipelines.

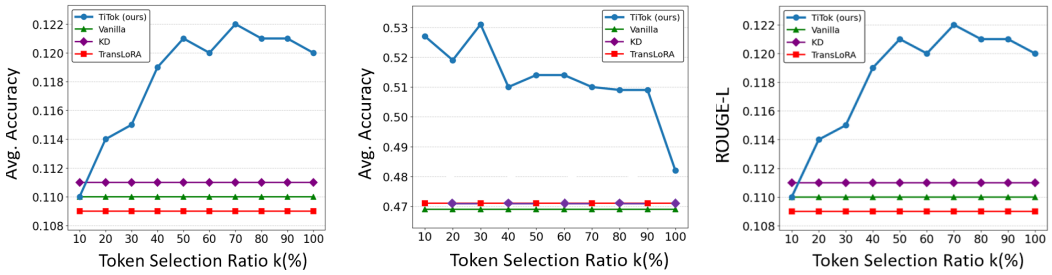
Collectively, these results provide strong evidence that TITOK not only excels in transfers within the same model family, but also demonstrates broad effectiveness in cross-model transfers, thereby highlighting its robustness across a wide spectrum of model scales, versions, and families.

4.3 ADDITIONAL ANALYSES

Ablation study. To validate the contribution of each component in our framework, we conduct additional experiments by selectively excluding the sample filtering and token selection mechanisms in Sec. 3.3. We report the average performance scores across all four transfer settings, and the results are presented in Table 2. When sample filtering is not applied (1st and 2nd rows), the training data for the target model is randomly sampled from the full synthetic dataset. Under this setting, applying only token selection (2nd row) performs noticeably better than the purely random baseline. This demonstrates the effectiveness of our token selection approach and empirically confirms that contrastive excess reliably identifies and selects the most informative tokens. Similarly, when only sample filtering is applied (1st and 3rd rows), the results improve over pure random sampling, indicating that selecting high-quality data is essential. This finding further underscores

Table 2: **Ablation study.** "Sample filtering" uses the mean token-level contrastive excess score to remove uninformative examples, while "Token selection" further refines the retained data by selectively keeping only the most informative tokens within each sample. Without sample filtering, data are randomly sampled. Results are reported as accuracy (Acc.) for BBH and MMLU, averaged across tasks. Meanwhile, we report ROUGE-1 (R-1) and ROUGE-L (R-L) for LaMP tasks.

Settings		BBH	MMLU	News Headline		Scholarly Title	
Sample filtering	**Token selection**	Acc.	Acc.	R-1	R-L	R-1	R-L
✗	✗	0.458	0.485	0.133	0.117	0.456	0.393
✗	✓	0.463	0.496	0.137	0.121	0.460	0.397
✓	✗	0.470	0.500	0.139	0.122	0.460	0.397
✓	✓	**0.483**	**0.501**	**0.142**	**0.125**	**0.464**	**0.403**



(a) BBH (accuracy) Mistral 7B → Mistral 7B

(b) BBH (accuracy) Llama2 7B → Llama3 8B

(c) News Headline (ROUGE-L) Llama2 7B → Llama3 8B

Figure 3: **Representative performance trends across $k\%$.** Among the three graphs, two come from the same task (BBH), while two share the same transfer setting (Llama2 7B → Llama3 8B).

that incorporating an effective filtering mechanism for synthetic data is essential. In our framework, contrastive excess serves this role by reliably selecting samples with richer and more informative signals, as demonstrated clearly by the empirical results Finally, the results show that combining both stages (4th row) achieves the best performance, confirming their complementary roles in filtering high-quality examples and selecting informative tokens for effective knowledge transfer.

Impact of query generation model. We now move on to examine how the choice of query generation model influences performance. Practically, synthetic queries can be generated either by the source model or by the target model, with the latter being the approach originally adopted in TransLoRA's pipeline (Wang et al., 2024). Figure 2 compares these two options across different transfer settings, with the reported scores averaged over all BBH tasks. Notably, we observe that using the source expert model (source backbone + LoRA) for query generation generally yields stronger performance than using the target model. One possible explanation is that when both the synthetic queries and the corresponding labels are generated by the same model, they remain closer to its training distribution. This alignment between queries and labels likely makes the supervision more coherent and accurate, thereby facilitating more effective transfer. These results suggest that maintaining distributional alignment between queries and labels is a key factor for improving knowledge transfer, thereby providing empirical justification for our design choice of using the source model as both the synthetic query and label generator.

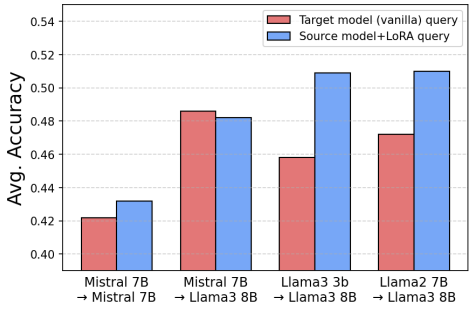


Figure 2: **Impact of query source.** Across transfer settings, using the source expert model to synthesize query generally yields better performance. The accuracy averaged over all BBH tasks are reported.

Effect of token selection ratio. We now proceed to explore the impact of token selection ratios ($k\%$) in our framework. Fig. 3 presents three representative curves: one task (BBH) under two

transfer settings, and another task (News Headline Generation) under the same transfer setting for comparison. For BBH with the same backbone on source and target (Mistral 7B→Mistral 7B), a moderate token selection ratio of 40-70% yields the best results. Very low ratios (10-20%) lead to underfitting, while a 100% ratio is ineffective as it applies no filtering at all. Interestingly, in a weak-to-strong transfer (Llama2 7B → Llama3 8B) for the same BBH task, performance improves as k% decreases. This suggests that selecting only the tokens with the highest excess loss effectively filters out the majority of noisy regions where the weaker model is uncertain. By discarding the majority of tokens where the weaker model lacks confidence, TITOK can significantly reduce negative transfer. This trend, however, reverses for the News Headline Generation task under the same weak-to-strong transfer setting. For this task, a larger k% is generally better, potentially because even a weaker model can provide valuable lexical and stylistic cues that are useful for personalization. Consequently, unlike reasoning-focused tasks such as BBH, which are sensitive to noisy supervision, personalization tasks like News Headline Generation benefit more broadly from the source model's outputs, even when the source model is relatively weaker than the target model.

Effectiveness of transfer through external data source. We further examine whether TITOK can transfer knowledge effectively in cases where synthetic data is not preferred, and thus external data is used as an alternative. To this end, we evaluate three alternative settings on LaMP tasks for Mistral 7B→Mistral 7B transfer setup: (1) using data from a randomly chosen different user, (2) mixing data from multiple users, and (3) transferring across tasks, where data of Scholarly Title Generation is used to train on News Headline Generation and vice versa (*i.e.*, out-of-distribution scenario). The results are presented in Table 3. Remarkably, TITOK consistently outperforms all baselines across these heterogeneous external settings. These findings demonstrate that TITOK is not restricted to synthetic data scenarios, but can also adapt effectively under external or user-provided data conditions. This highlights the flexibility of TITOK for practical deployment in diverse real-world application scenarios and further underscores its adaptability across different data conditions, as it is effective even when external data is used as an alternative to synthetic data.

Table 3: **Transfer using external data.** ROUGE-1 (R-1) and ROUGE-L (R-L) on News Headline and Scholarly Title Generation under three data settings for Mistral 7B→Mistral 7B: (1) other-user, (2) mixed-user, (3) cross-task. Best scores are in **bold**.

Data Setting	Method	News Headline		Scholarly Title	
		R-1	R-L	R-1	R-L
Other-user	Vanilla	0.117	0.101	0.381	0.311
	KD	0.127	0.113	0.405	0.333
	TiTok (ours)	**0.151**	**0.136**	**0.481**	**0.425**
Mixed-user	Vanilla	0.117	0.101	0.381	0.311
	MinED	0.124	0.110	0.409	0.337
	TiTok (ours)	**0.151**	**0.137**	**0.480**	**0.422**
Cross-task	Vanilla	0.117	0.101	0.381	0.311
	MinED	0.118	0.106	0.403	0.331
	TiTok (ours)	**0.133**	**0.120**	**0.450**	**0.383**

5 CONCLUSION

In this paper, we propose TITOK, an efficient framework that transfers LoRA knowledge from a source model to a target model by training only on a selectively chosen set of highly informative tokens. At its core, TITOK leverages token-level signals to distill task-relevant information from the source adapter, rather than relying on the entire token sequence. By focusing supervision on the regions where the adapter contributes most, TITOK achieves stronger and more targeted knowledge transfer. With this simple yet effective design, TITOK proves robust across tasks and consistently surpasses existing baselines, making it a practical solution for efficient knowledge transfer.

Limitations and Future Directions While TITOK has shown robustness and clear advantages over existing methods, there remain opportunities to refine and extend the framework. Firstly, although TITOK is designed to minimize dependence on original data, it still requires a small number of seed examples to generate synthetic data via prompting. Nevertheless, this reliance is modest, as we avoid using full datasets. In addition, our experiments confirm that external data can also serve as an effective alternative, reinforcing the flexibility of the approach. Regarding token selection, TITOK currently applies a fixed threshold to determine which tokens are retained. While this simple design is effective and stable across tasks, future work could explore more adaptive or data-driven thresholding strategies to further enhance efficiency without compromising robustness.

Ethics Statement

We conduct this research in full accordance with established ethical standards. For our experiments we rely exclusively on publicly available datasets such as LaMP, MMLU, and BBH and use them strictly in accordance with their intended purpose for academic research. For the LaMP tasks, which involve user data, our TiTok framework aligns with ethical considerations by minimizing data dependence. It does not store or expose raw user data and only updates a small set of task and user-specific parameters. This helps safeguard privacy while enabling efficient knowledge transfer.

Reproducibility Statement

We provide a comprehensive description of our implementation in Section 4, including pipeline configurations, hyperparameters, models, datasets, and evaluation metrics. The source code for our implementation and experiments will be made publicly available in a repository upon publication.

References

Rambod Azimi, Rishav Rishav, Marek Teichmann, and Samira Ebrahimi Kahou. Kd-lora: A hybrid approach to efficient fine-tuning with lora and knowledge distillation. *arXiv preprint arXiv:2410.20777*, 2024.

Melis Ilayda Bal, Volkan Cevher, and Michael Muehlebach. Eslm: Risk-averse selective language modeling for efficient pretraining, 2025. URL https://arxiv.org/abs/2505.19893.

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, 2020. URL https://arxiv.org/abs/2005.14165.

Hyungjune Bu, Chanjoo Jung, Minjae Kang, and Jaehyung Kim. Personalized llm decoding via contrasting personal preference, 2025. URL https://arxiv.org/abs/2506.12109.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations (ICLR)*, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. Train no evil: Selective masking for task-guided pre-training. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6966–6974, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.566. URL https://aclanthology.org/2020.emnlp-main.566/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Seungone Kim, Juyoung Suk, Xiang Yue, Vijay Viswanathan, Seongyun Lee, Yizhong Wang, Kiril Gashteovski, Carolin Lawrence, Sean Welleck, and Graham Neubig. Evaluating language models as synthetic data generators. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models, 2021. URL https://arxiv.org/abs/2003.02245.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations, 2023. URL https://arxiv.org/abs/2310.07849.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need, 2025. URL https://arxiv.org/abs/2404.07965.

He Liu, Yikai Wang, Huaping Liu, Fuchun Sun, and Anbang Yao. Small scale data-free knowledge distillation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney,

Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization, 2024. URL https://arxiv.org/abs/2304.11406.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models, 2024. URL https://arxiv.org/abs/2401.10491.

Runqian Wang, Soumya Ghosh, David Cox, Diego Antognini, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. Trans-lora: towards data-free transferable parameter efficient finetuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. When search engine services meet large language models: Visions and challenges, 2024. URL https://arxiv.org/abs/2407.00128.

Seo Yeongbin, Dongha Lee, and Jinyoung Yeo. Train-attention: Meta-learning where to focus in continual knowledge learning. *Advances in Neural Information Processing Systems*, 37:58284–58308, 2024.

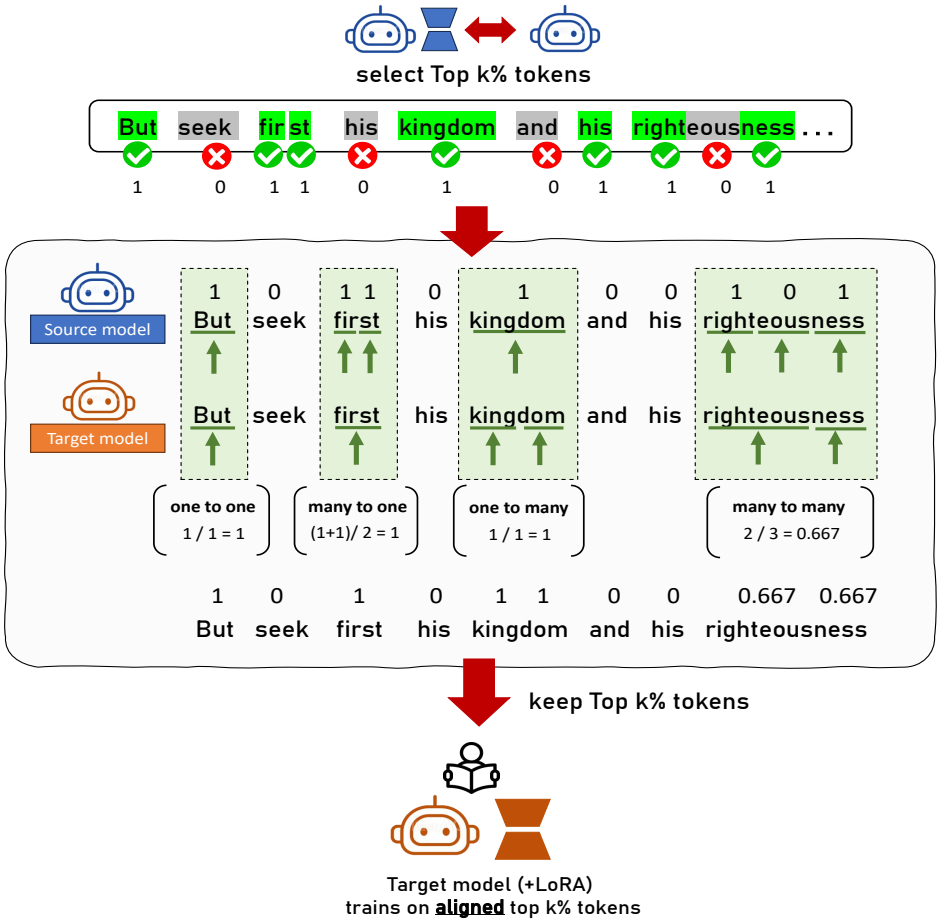


Figure 4: Overview of **TITOK's tokenizer alignment algorithm.** The algorithm handles cases where the source and target models use different tokenizers. The binary mask scores assigned by the source model are averaged within aligned spans and propagated to target tokens, producing fractional scores that guide top-$k\%$ token selection for training the target model's LoRA adapter.

A TOKENIZER ALIGNMENT ALGORITHM

When the source and target models use different tokenizers, direct token-level transfer is not possible. To address this, we implement a dual-pointer alignment procedure. The algorithm maintains two pointers, one for the source tokens and one for the target tokens. At each step, the source pointer advances by one token, accumulating a decoded segment, while the target pointer incrementally extends its own segment until the normalized texts match. Once a match is found, the corresponding spans are recorded as an alignment, and the target pointer jumps forward to that position. After alignment, we apply masking rules to propagate the source binary mask scores (values that indicate whether a token should be kept or discarded) to the target tokens. Specifically:

1) **One-to-One**: binary mask score is directly copied.
2) **One-to-Many**: score is replicated across all aligned target tokens.
3) **Many-to-One**: averaged scores of multiple source tokens are assigned to the target token.
4) **Many-to-Many**: the averaged score of the source tokens is assigned to the corresponding aligned target tokens.

This process yields fractional mask scores that capture the relative importance of each target token. Finally, we keep only the top-$k\%$ of tokens according to these scores, producing a final binary selection mask over the target tokens that preserves the most informative regions while discarding less relevant ones. An overview of this tokenizer alignnment algorithm is presented in Figure 4.

B Details of Datasets

B.1 Big-Bench Hard (BBH)

Big-Bench Hard (BBH) (Suzgun et al., 2022) is designed as a rigorous benchmark for evaluating model performance on challenging reasoning problems, including multi-step logical reasoning, symbolic manipulation, and commonsense inference. The tasks are formatted as multiple choice questions. Since BBH is originally a test-only benchmark, we split 90% of the data for training the source expert model and reserved 10% for evaluation. The 27 BBH tasks are categorized in Table 4.

Table 4: Categorization of the 27 Big-Bench Hard (BBH) tasks.

Category	**Tasks**
Logical Reasoning	boolean_expressions, causal_judgement, date_understanding, disambiguation_qa, dyck_languages, formal_fallacies, logical_deduction_three_objects, logical_deduction_five_objects, logical_deduction_seven_objects, temporal_sequences, tracking_shuffled_objects_three_objects, tracking_shuffled_objects_five_objects, tracking_shuffled_objects_seven_objects, web_of_lies
Linguistic	hyperbaton, ruin_names, salient_translation_error_detection, snarks, word_sorting
Mathematical / Symbolic	geometric_shapes, multistep_arithmetic_two, object_counting, reasoning_about_colored_objects
Applied / Knowledge	movie_recommendation, navigate, penguins_in_a_table, sports_understanding

Table 5: Categorization of the 57 MMLU tasks.

Category	**Tasks**
STEM (29)	abstract_algebra, anatomy, astronomy, college_biology, college_chemistry, college_computer_science, college_mathematics, college_medicine, college_physics, computer_security, conceptual_physics, electrical_engineering, elementary_mathematics, formal_logic, high_school_biology, high_school_chemistry, high_school_computer_science, high_school_mathematics, high_school_physics, high_school_statistics, machine_learning, medical_genetics, nutrition, professional_medicine, professional_psychology, virology, clinical_knowledge, human_aging, human_sexuality
Humanities (16)	business_ethics, formal_fallacies, jurisprudence, logical_fallacies, philosophy, prehistory, world_religions, moral_disputes, moral_scenarios, professional_law, professional_accounting, high_school_world_history, high_school_us_history, high_school_european_history, global_facts, security_studies
Social Sciences (10)	econometrics, high_school_macroeconomics, high_school_microeconomics, management, marketing, public_relations, sociology, us_foreign_policy, high_school_geography, high_school_government_and_politics
Other (2)	miscellaneous, global_facts

B.2 Massive Multitask Language Understanding (MMLU)

Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) is a comprehensive benchmark for evaluating model performance across a broad range of knowledge intensive tasks. The benchmark consists of multiple choice questions and, similarly to BBH, we also apply a 90%/10% split of the original test-only data. All the 57 subtasks are categorized in Table 5.

B.3 LaMP tasks

We utilize the LaMP benchmark to evaluate whether TiTok is also effective in the personalization setting. Among the LaMP tasks, we focus on the two text generation tasks that are suitable, accessible, and evaluable in our setting:

- **News Headline Generation (LaMP 4).** Given an author profile consisting of previously written headlines, the model is asked to generate a headline for a new news article. The task evaluates if the model can adapt its output to reflect the author's characteristic style in journalistic writing.
- **Scholarly Title Generation (LaMP 5).** Using an author profile built from prior titles of academic publications, the model generates a title for a new given abstract. The task assesses the model's ability to capture and reproduce distinctive conventions of academic writing.

C Baseline Details

To evaluate the effectiveness of TiTok, we compare against several baselines as follows:

C.1 Vanilla

The vanilla baseline corresponds to the standard target base model without any additional training or knowledge transfer from the source model. This setup reflects the performance of the target model in its raw initialized state and serves as a lower bound for evaluating transfer methods. Intuitively, achieving performance that surpasses this baseline provides clear evidence that the target model has successfully acquired and internalized knowledge transferred from the source model.

C.2 Knowledge Distillation (KD) (+MinED)

The knowledge distillation (KD) (Hinton et al., 2015; Azimi et al., 2024) baseline trains the student model to mimic the teacher model's output distribution. Specifically, the loss is a weighted sum of the cross entropy objective and the KL divergence between the teacher and student distributions. In our setup, the KD experiments are conducted using the TransLoRA filtered synthetic datasets.

For cases where the source and target models use mismatched tokenizers when performing KD, we apply the Minimal Edit Distance (MinED) alignment method (Wan et al., 2024). In particular, MinED matches tokens across different vocabularies by minimizing the number of character level edits. For example, it can align "get" with "gets," "color" with "colour," or "analysis" with "analyses." This tokenizer alignment approach avoids degeneracy issues in token alignment when conducting KD, though it is different from our dual-pointer based alignment algorithm.

C.3 TransLoRA

The TransLoRA baseline (Wang et al., 2024) transfers LoRA knowledge by generating synthetic data. In this method, the vanilla target model synthesizes queries, and the source model with its LoRA adapter provides the corresponding labels. Subsequently, a discriminator, separately trained with the source model as its base, is applied to filter the synthetic data used for fine-tuning the target model's LoRA adapter. For the sake of consistency and fair comparison,we adopt the same hyperparameter settings from our experiments when applying the TransLoRA baseline.

Table 6: **Comparison with a few-shot KD baseline using five seed prompts.** BBH and MMLU are reported as the average accuracy across tasks, while News Headline and Scholarly Title Generation tasks are evaluated using ROUGE-1 (R-1) and ROUGE-L (R-L). *KD (5-shot)* denotes knowledge distillation performed with only five few-shot examples. Best scores are in **bold**.

Transfer	Method	BBH	MMLU	News Headline		Scholarly Title	
		Acc.	Acc.	R-1	R-L	R-1	R-L
Mistral 7B → Mistral 7B	Vanilla	0.397	0.557	0.117	0.101	0.381	0.311
	KD (5-shot)	0.402	0.558	0.118	0.104	0.383	0.312
	TiTok (ours)	**0.432**	**0.563**	**0.160**	**0.142**	**0.473**	**0.414**
Mistral 7B → Llama 8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD (5-shot)	0.470	0.477	0.126	0.111	0.446	0.379
	TiTok (ours)	**0.482**	**0.488**	**0.140**	**0.124**	**0.464**	**0.403**
Llama3B → Llama8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD (5-shot)	0.470	**0.479**	0.126	0.111	0.446	0.379
	TiTok (ours)	**0.509**	0.475	**0.127**	**0.113**	**0.457**	**0.392**
Llama2 7B → Llama3 8B	Vanilla	0.469	0.469	0.125	0.110	0.444	0.378
	KD (5-shot)	0.470	0.477	0.126	0.111	0.446	0.380
	TiTok (ours)	**0.510**	**0.479**	**0.140**	**0.122**	**0.461**	**0.404**

D USAGE OF AI ASSISTANTS

AI assistants are minimally used in this work, restricted solely to language refinement. such as grammar correction, punctuation, and sentence structure. All research ideas, methodologies, and analyses are the original contributions of the authors. Thus, the use of AI is confined to editorial support and did not influence the originality or intellectual contributions of the work.

E COMPARISON WITH ADDITIONAL BASELINE

Here, we introduce an additional baseline that leverages few-shot supervision. In practice, our synthetic data generation process is seeded with a small set of prompts, which can be regarded as few-shot data. To be thorough, we therefore establish a baseline trained only on these five seed prompts, referred to as KD (5-shot), so that our evaluation of TITOK also considers minimal few-shot supervision. The results of this additional baseline are presented in Table 6.

Across all transfer settings, KD on only 5-shot samples provides only marginal improvements over the vanilla model, with average gains of 0.9% on reasoning tasks (BBH and MMLU) and 0.6–1.2% on personalization tasks (News Headline and Scholarly Title Generation). In contrast, TITOK delivers substantial improvements over this baseline, achieving 4.7% and 2.1% average gains on reasoning tasks and 8.9–16.6% average gains on personalization tasks. Winning over KD with only 5 few-shot samples shows that TITOK is not simply the result of a few-shot effect. Instead, the five seed prompts act only as a starting point, which our framework expands into richer and more effective synthetic training signals. This finding confirms that the true strength of TITOK lies in how it effectively leverages limited supervision rather than in the few-shot data itself.

F SYNTHETIC DATA GENERATION WITH 2× POOL AND TOP-M SELECTION

In this section, we provide further details regarding the number of synthetic data samples used. In generating synthetic data, we first provide five samples from the original training set as few-shot exemplars to guide the model toward producing outputs in the desired style and format. For each task, we initially create a synthetic pool containing twice the number of examples used in the source model's training. This pool is then filtered using contrastive excess scores, after which we retain only the top M samples, where M equals the size of the source training set. To be specific, we set

$M = 250$ for BBH, $M = 90$ for MMLU, and $M = 200$ for LaMP tasks. This procedure ensures that the target model's LoRA adapter is trained on a dataset comparable in scale to that of the source model, while selective filtering enhances the overall quality of the retained data.

G ROUGE-L Filtering for Diverse Synthetic Queries

We now proceed to provide the task lists for which we did not apply ROUGE-L filtering when generating diverse synthetic queries. In general, we use ROUGE-L filtering to encourage diversity in queries, but for the tasks listed in Table 7, we only applied simple deduplication. In the case of BBH, tasks such as *boolean_expressions* or *temporal_sequences* already follow highly restricted and repetitive patterns, making high ROUGE-L scores inevitable. For MMLU, the only tasks without ROUGE-L filtering are the history-related subjects. This is potentially because history questions often require long passages that overlap in vocabulary, phrasing, or factual references (*e.g.*, recurring names, dates, or events). Applying strict ROUGE-L filtering in such cases would make it difficult to generate the required number of synthetic queries. For all remaining BBH and MMLU tasks, as well as all LaMP tasks, we applied a ROUGE-L threshold of 0.7 to encourage diversity while still preserving task fidelity. Table 7 provides the tasks for which ROUGE-L filtering is not applied.

Table 7: Tasks without ROUGE-L filtering.

Category	**Tasks**
BBH	bbh_boolean_expressions, bbh_date_understanding, bbh_disambiguation_qa, bbh_geometric_shapes, bbh_logical_deduction_three_objects, bbh_multistep_arithmetic_two, bbh_navigate, bbh_object_counting, bbh_penguins_in_a_table, bbh_reasoning_about_colored_objects, bbh_salient_translation_error_detection, bbh_snarks, bbh_temporal_sequences, bbh_tracking_shuffled_objects_three_objects, bbh_web_of_lies
MMLU	high_school_world_history, high_school_us_history, high_school_european_history

H Prompts for Synthetic Query Generation

In this section, we present the prompts used for generating synthetic queries. Each prompt includes five examples, which correspond to the seed prompts taken from the original data. For BBH, which consists of multiple subtasks, we specify task-specific instructions and formatting rules so that the generated queries and labels are structured according to the expected format. Each BBH subtask is paired with its corresponding rules and output format to ensure consistency across the benchmark, and Table 8 illustrates the case of *boolean_expressions* as a representative example.

In contrast, for MMLU a single unified prompt format is sufficient for most subjects, and the model reliably generates queries that follow the expected structure. The exception is the history-related tasks, namely, *high_school_us_history*, *high_school_world_history*, and *high_school_european_history*. For these tasks, it is observed that more specific prompting is required to obtain better generations. The general MMLU prompt is shown in Table 9, whereas the history-related tasks make use of the specialized prompt provided in Table 10.

Finally, for LaMP tasks, we apply the same query template uniformly to each user. The detailed prompt templates for LaMP 4 and 5 are provided in Tables 11 and 12 respectively.

Table 8: Synthetic query prompt for the BBH *boolean_expressions* task. Each BBH subtask requires task-specific instructions and formatting rules to ensure that generated queries and labels follow the expected format. We show the *boolean_expressions* task here as a representative example.

Synthetic query generation prompt for BBH (*boolean_expressions*) task.

System
You are an expert task generator for boolean_expressions tasks. Generate boolean expression evaluation tasks ending with `` ` is ` `` – SINGLE LINE ONLY.

CRITICAL FORMAT REQUIREMENTS:
- Follow the EXACT format structure shown in the examples.
- ABSOLUTELY CRITICAL: Generate **ONLY ONE LINE**, ONE TASK per response.
- NO multiple lines, NO newlines (`\n`), NO multiple expressions.
- Must end with `` ` is ` ``.
- Example of **INVALID** output: `True or False is\n\n False and True is`
- Example of **VALID** output: `True or False or not False and ( True or False ) is`

Generate diverse content but maintain the exact same format structure. Only output the task input, not the solution.

User
Generate new `boolean_expressions` tasks following these exact format examples:
Example 1: [boolean expression ending with `` ` is ` ``]
Example 2: [boolean expression ending with `` ` is ` ``]
Example 3: [boolean expression ending with `` ` is ` ``]
Example 4: [boolean expression ending with `` ` is ` ``]
Example 5: [boolean expression ending with `` ` is ` ``]

CRITICAL: Generate ONLY ONE LINE, exactly like the examples above.

Generate a new task following the exact format:

Table 9: Synthetic query prompt for the MMLU. Applied to all tasks except for history related tasks.

Synthetic query generation prompt for MMLU (general).
Example 1: Question: [question snippet] 1. [choice snippet] 2. [choice snippet] 3. [choice snippet] 4. [choice snippet] Answer: [1–4] Example 2: Question: [question snippet] 1. [choice snippet] 2. [choice snippet] 3. [choice snippet] 4. [choice snippet] Answer: [1–4] Example 3: Question: [question snippet] 1. [choice snippet] 2. [choice snippet] 3. [choice snippet] 4. [choice snippet] Answer: [1–4] Example 4: Question: [question snippet] 1. [choice snippet] 2. [choice snippet] 3. [choice snippet] 4. [choice snippet] Answer: [1–4] Example 5: Question: [question snippet] 1. [choice snippet] 2. [choice snippet] 3. [choice snippet] 4. [choice snippet] Answer: [1–4] Example 6:

Table 10: Synthetic query prompt for the MMLU history tasks only. The history tasks are *high_school_us_history*, *high_school_world_history*, and *high_school_european_history*. These history tasks required more specific prompting to obtain better results.

Synthetic query generation prompt for MMLU (history related tasks).

Generate [SUBJECT] multiple choice questions following these examples:

Example 1:
Question: [history question snippet]
1. [choice snippet]
2. [choice snippet]
3. [choice snippet]
4. [choice snippet]
Answer: [1–4]

Example 2:
Question: [history question snippet]
1. [choice snippet]
2. [choice snippet]
3. [choice snippet]
4. [choice snippet]
Answer: [1–4]

Example 3:
Question: [history question snippet]
1. [choice snippet]
2. [choice snippet]
3. [choice snippet]
4. [choice snippet]
Answer: [1–4]

Example 4:
Question: [history question snippet]
1. [choice snippet]
2. [choice snippet]
3. [choice snippet]
4. [choice snippet]
Answer: [1–4]

Example 5:
Question: [history question snippet]
1. [choice snippet]
2. [choice snippet]
3. [choice snippet]
4. [choice snippet]
Answer: [1–4]

Now generate a new [SUBJECT] question following the same format:
Question:

Table 11: Synthetic query prompt for the News Headline Generation task.

Synthetic query generation prompt for News Headline Generation
You are a news text generator. Generate diverse news article texts that could be used to create headlines. Only output the raw news text content, not headlines or queries. Example 1: [article snippet] Example 2: [article snippet] Example 3: [article snippet] Example 4: [article snippet] Example 5: [article snippet] Example 6:

Table 12: Synthetic query prompt for the Scholarly Title Generation task.

Synthetic query generation prompt for Scholarly Title Generation
You are a scholarly abstract generator. Generate diverse ONE paragraph abstracts that ask for creating paper titles from abstracts. The abstract should be ONE paragraph only. Only output the raw abstract text, not the actual titles. Example 1: Create a title for this research paper: Abstract: "[abstract snippet]" Title: "[title]" Example 2: Create a title for this research paper: Abstract: "[abstract snippet]" Title: "[title]" Example 3: Create a title for this research paper: Abstract: "[abstract snippet]" Title: "[title]" Example 4: Create a title for this research paper: Abstract: "[abstract snippet]" Title: "[title]" Example 5: Create a title for this research paper: Abstract: "[abstract snippet]" Title: "[title]" Write a research paper abstract as a single paragraph containing at least 3 sentences. Example 6: