LLM2LLM: PARAMETER-EFFICIENT ADAPTATION VIA TRANSFERRED EXPERTS FROM MOE LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Stronger language models acquire rich knowledge through expensive pretraining, but transferring it typically requires learning from teacher-generated supervision. We propose **LLM2llm**, a framework that extracts reusable expert components from a stronger source Mixture-of-Experts model for direct inheritance by target pretrained model. LLM2llm identifies *cross-domain generalist experts* using multi-domain routing statistics and condenses them through low-rank subspace extraction and score-weighted fusion. This one-time extraction requires no additional training, and the resulting experts can be reused across multiple downstream tasks. Injected as residual branches, only the transferred experts are updated during downstream supervised fine-tuning; the original target parameters remain frozen, and the source model is no longer required. Across two heterogeneous source–target settings, LLM2llm achieves higher average performance than Full Fine-Tuning and representative parameter-efficient baselines on language understanding and dialogue generation tasks. Ablations support the effectiveness of expert selection, condensation, and transferred initialization. These results demonstrate the potential of directly inheriting reusable model components to improve downstream learning efficiently. Code is available at `https://anonymous.4open.science/r/LLM2llm-4E51/`.

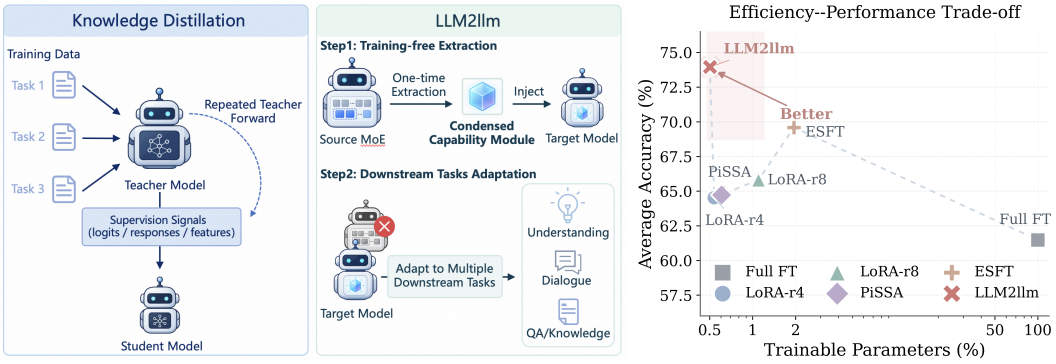
1 INTRODUCTION

Large Language Models (LLMs) have achieved remarkable capabilities across a wide range of tasks, with performance generally improving as model scale and pretraining resources increase (Grattafiori et al., 2024; Yang et al., 2025). However, stronger models often incur substantial computational, memory, and deployment costs, limiting their practicality in resource-constrained scenarios. Smaller pretrained models provide a more efficient alternative, but typically exhibit a considerable capability gap relative to their larger counterparts. This naturally motivates an important question: how can capabilities already acquired by stronger models be effectively transferred to smaller and more deployable models? Knowledge distillation provides a widely adopted paradigm for transferring capabilities from a stronger teacher model to a smaller student model (Gu et al., 2024; Team et al., 2025). Typically, the teacher produces output distributions (Gu et al., 2024; Agarwal et al., 2024), generated responses (Guo et al., 2025), or intermediate representations (Jiao et al., 2020) that serve as supervision for training the student. In this paradigm, the knowledge of the teacher model is converted into supervision from which the student learns through optimization.

Despite its effectiveness, distillation imposes substantial resource demands: hosting a large teacher requires considerable memory, while repeated teacher forward passes consume computation and time, whether performed during student training or beforehand to generate supervision (Gu et al., 2024; Agarwal et al., 2024). Feature-based distillation introduces further complexity, as mismatched hidden dimensions prevent direct feature matching without additional alignment (Romero et al., 2014; Jiao et al., 2020). More fundamentally, distillation accesses teacher knowledge through output signals or intermediate features, requiring the student to reconstruct capabilities through optimization instead of directly reusing the parameter structures in which that knowledge is encoded.

This motivates a complementary perspective on model-to-model capability transfer: ***can knowledge already acquired by a stronger model be extracted into reusable components and directly inherited by another pretrained model to improve its downstream learning and attainable perfor-***

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Knowledge transfer paradigms. (b) Accuracy vs. trainable parameters.

Figure 1: (a) Unlike knowledge distillation through teacher-generated supervision, LLM2llm extracts reusable expert modules from a source MoE for direct inheritance and downstream adaptation by a pretrained target model. (b) Average accuracy across language understanding benchmarks versus the percentage of trainable parameters, with Qwen3.5-122B as the source and OLMoE-7B as the target for LLM2llm.

mance? Such a mechanism would allow pretrained models to build on knowledge-bearing components learned by stronger models, enabling efficient knowledge reuse for downstream adaptation.

Mixture-of-Experts (MoE) models provide a particularly suitable foundation for this idea. MoE architectures underpin several leading large-scale language models, making them practically relevant sources of advanced capabilities (Jiang et al., 2024; Dai et al., 2024; Liu et al., 2024a; Zeng et al., 2025). Their explicitly routed experts also exhibit domain-dependent activation patterns, providing identifiable modules for analyzing and extracting knowledge relevant to different tasks (Muennighoff et al., 2025). While some experts specialize in particular domains, others remain active across diverse data, suggesting that they may capture useful and generalizable knowledge beyond individual domains. We refer to the latter as *cross-domain generalist experts* and investigate whether their parameters can provide reusable knowledge for downstream adaptation in other pretrained models.

Based on this insight, we propose **LLM2llm**, a cross-model capability transfer framework that extracts reusable expert knowledge from a stronger source model to enhance downstream learning in a pretrained target model. As shown in Figure 2, LLM2llm first collects expert routing statistics on data from diverse domains to select cross-domain generalist experts with broad knowledge coverage. It then refines and consolidates the selected experts through low-rank subspace extraction and score-weighted fusion, yielding one condensed expert per selected source layer for injection into the target model. Both stages are *training-free*. The experts are extracted once and can be reused for adaptation across multiple downstream tasks, as illustrated on the right side of Figure 1a.

The condensed experts are injected as residual branches alongside selected target feed-forward modules, with fixed projections bridging hidden dimensions (Figure 2). During downstream supervised fine-tuning (SFT), all original target-model parameters *remain frozen*; only the injected source experts are updated. This enables efficient reuse of source-model knowledge to improve downstream learning, without executing the source model or requiring distillation supervision during adaptation.

We evaluate LLM2llm under two heterogeneous source–target settings, Qwen3.5-122B-A10B (Qwen Team, 2026) to OLMoE-7B (Muennighoff et al., 2025) and Qwen3.5-35B-A3B (Qwen Team, 2026) to LLaMA2-7B (Touvron et al., 2023), covering different model scales and backbone architectures. Across a broad suite of language understanding and dialogue generation tasks, LLM2llm consistently outperforms Full Fine-Tuning (Full FT), LoRA (Hu et al., 2021), PiSSA (Meng et al., 2024), and the target-native expert tuning method ESFT (Wang et al., 2024), while updating only **0.54%** and **0.19%** of the target parameters in the two settings, respectively. Extensive ablations further validate the effectiveness of generalist expert selection, low-rank condensation, and cross-model expert injection. These results demonstrate the feasibility of transferring and inheriting knowledge encoded in stronger language models to extend the capabilities attainable by target models. More broadly, LLM2llm provides a new direction for cross-model knowledge reuse, where knowledge acquired by stronger models can be transferred to enhance learning and down-

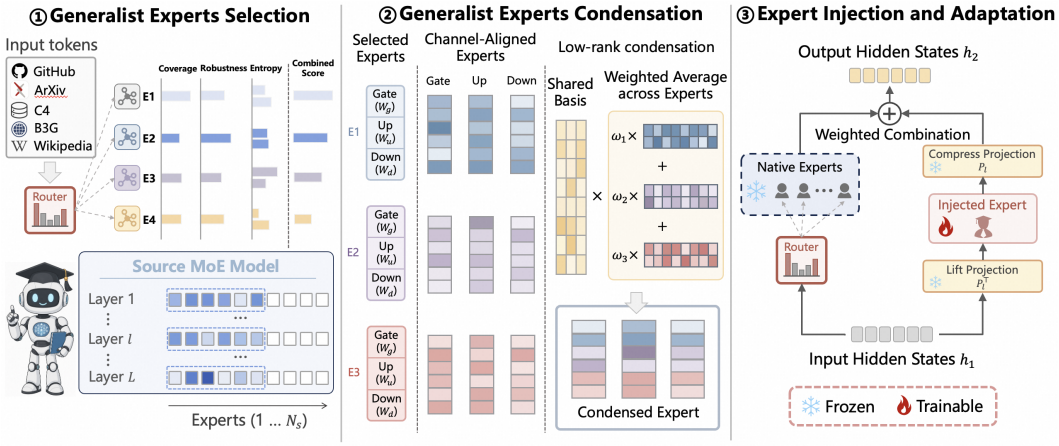


Figure 2: Overview of **LLM2llm**. Cross-domain generalist experts are first identified from a stronger source MoE, then channel-aligned and condensed through shared low-rank subspace extraction and score-weighted fusion. The resulting condensed generalist experts are injected into selected target layers as trainable residual branches for downstream adaptation. The target model in LLM2llm can adopt either a dense or MoE architecture; for illustration, we use an MoE target model in the figure.

stream performance of other models in specialized domains. Our contributions are summarized as follows:

- We propose LLM2llm, a cross-model knowledge transfer framework that enables pretrained models to directly inherit reusable expert components from a stronger source model, enhancing downstream learning by updating only the injected experts while keeping the original target parameters frozen.
- We develop a training-free expert selection and condensation method that identifies cross-domain generalist experts and consolidates their knowledge into reusable modules. These modules are extracted once and reused for adaptation across multiple downstream tasks.
- Experiments on two heterogeneous source–target pairs show that LLM2llm achieves higher average performance than Full FT, LoRA, PiSSA, and ESFT on language understanding and generation tasks while updating only a small fraction of parameters, supporting efficient cross-model knowledge reuse.

2 Related Work

Reusable Knowledge in Models. Prior work has explored identifying and reusing knowledge encoded in pretrained source models through parameter importance estimation, knowledge inheritance, and knowledge-aware initialization (Molchanov et al., 2019; Frankle & Carbin, 2018; Liu et al., 2026a; Shi et al., 2026; Liu et al., 2026b). Importance-based methods often identify dispersed parameters or coarse-grained components, making the resulting knowledge difficult to reuse as structured modules. Knowledge inheritance methods instead extract generalizable representations to initialize target models, but typically operate before or during pretraining and require substantial subsequent training. MoE architectures provide a more explicit modular structure, in which experts can serve as natural units of reusable knowledge. However, how expert knowledge from a stronger source MoE can be transferred to efficiently adapt an already pretrained target model remains underexplored.

Parameter-Efficient Fine-Tuning PEFT adapts pretrained models by updating only a small fraction of parameters while keeping the backbone frozen. Representative methods introduce lightweight task-specific modules, with subsequent variants improving rank allocation, parameter sharing, weight decomposition, or initialization (Houlsby et al., 2019; Hu et al., 2021; Zhang et al., 2023; Kopiczko et al., 2024; Liu et al., 2024b; Meng et al., 2024). For MoE models, ESFT selectively updates target experts relevant to downstream tasks (Wang et al., 2024). Despite their effec-

tiveness, these methods remain largely target-centric: their trainable parameters are either randomly initialized or derived solely from the target model, without leveraging richer expert knowledge from a stronger source MoE. In contrast, LLM2llm transfers condensed expert knowledge from a source MoE into a pretrained target model as parameter-efficient adaptation modules.

3 LLM2LLM

To enable pretrained models to inherit capabilities already acquired by a stronger MoE, we extract source expert knowledge into reusable model components. We introduce LLM2llm, which incorporates these components directly into a pretrained target model as trainable adaptation modules. As illustrated in Figure 2, the framework comprises three stages: cross-domain generalist expert selection, expert condensation through channel alignment and low-rank fusion, and cross-model expert injection for downstream adaptation.

Let $\mathcal{M}_s$ and $\mathcal{M}_t$ denote the source MoE and target model, respectively. The source contains L_s MoE layers with N_s routed experts per layer, and $E^s_{\ell,e}$ denotes expert e in source layer ℓ. Our goal is to condense selected experts into a transferable module $\widehat{E}^s_\ell$ while keeping the original target parameters frozen.

3.1 Cross-Domain Generalist Expert Selection

To identify source components with potential for reuse across downstream tasks, we favor experts whose routing activity is both substantial and broadly distributed across domains. We treat these patterns as an empirical indicator of cross-domain generality and use them to select candidate carriers of reusable knowledge, as shown in the first stage of Figure 2.

We collect routing statistics on calibration data from domains $\mathcal{C} = \{c_1, \ldots, c_D\}$. For token t from domain c, let $\mathcal{T}_{c,\ell,t}$ denote the experts selected by the native top-k router in source layer ℓ. The normalized routing activity of expert e is

$$a_{c,\ell,e} = \frac{\sum_t \mathbb{I}[e \in \mathcal{T}_{c,\ell,t}]}{\sum_{e'=1}^{N_s} \sum_t \mathbb{I}[e' \in \mathcal{T}_{c,\ell,t}]}, \tag{1}$$

which measures its share of routing selections within that domain and layer. To compare relative activity across domains, we apply z-score normalization to $a_{c,\ell,e}$ across experts separately for each domain and layer, obtaining $z_{c,\ell,e}$.

We characterize generality using three complementary statistics. ***Coverage***, $\mathrm{Cov}_{\ell,e}$, is the fraction of domains where $z_{c,\ell,e}$ exceeds an activity threshold τ. ***Robustness*** is the lower quartile of $z_{c,\ell,e}$ across domains; we apply min–max normalization across experts within each layer to obtain $\widetilde{\mathrm{P25}}_{\ell,e}$. ***Entropy***, $\mathrm{Ent}_{\ell,e}$, measures the evenness of domain usage: we normalize $a_{c,\ell,e}$ across domains to form $\pi_{c,\ell,e} = \frac{a_{c,\ell,e}}{\sum_{c'} a_{c',\ell,e}}$ and compute its Shannon entropy divided by $\log D$. Coverage and entropy already lie in $[0, 1]$. The generalist score is

$$s_{\ell,e} = \lambda_1 \mathrm{Cov}_{\ell,e} + \lambda_2 \widetilde{\mathrm{P25}}_{\ell,e} + \lambda_3 \mathrm{Ent}_{\ell,e}, \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{3} \lambda_i = 1. \tag{2}$$

To ensure sufficient domain coverage, we retain experts satisfying $\mathrm{Cov}_{\ell,e} \geq \gamma$, where γ is a minimum coverage requirement, and select up to m of them with the highest scores. We denote this set by $\mathcal{S}_\ell$. The score determines which experts are selected, whereas m controls how many contribute to condensation. Together, these statistics favor broad and stable use while reducing the influence of isolated activity peaks in individual domains.

3.2 Cross-Domain Generalist Expert Condensation

Although the selected experts are broadly active across domains, their activation profiles differ, and their parameters may still encode domain-specific knowledge. We therefore aim to further distill their shared knowledge while suppressing expert-specific variation, forming a compact component

for cross-domain reuse. To this end, we first align intermediate channels, then extract a shared parameter subspace and perform score-weighted fusion within each source layer, as illustrated in the second stage of Figure 2.

Each selected expert has gate, up, and down projections $(W^g_{\ell,e}, W^u_{\ell,e}, W^d_{\ell,e})$, with gate and up matrices in $\mathbb{R}^{h_s \times d_s}$ and the down matrix in $\mathbb{R}^{d_s \times h_s}$, where d_s and h_s are the source hidden and expert intermediate dimensions. Since intermediate channels are permutation-equivalent, similar features may occupy different positions. We use the highest-scoring selected expert e^{ref}_ℓ as the reference and compare its channel i with channel j of expert e through

$$\begin{aligned}\text{Sim}_{\ell,e}(i,j) = \alpha_g \cos\Big(W^g_{\ell,e^{\text{ref}}_\ell}[i,:], W^g_{\ell,e}[j,:]\Big) \\ + \alpha_u \cos\Big(W^u_{\ell,e^{\text{ref}}_\ell}[i,:], W^u_{\ell,e}[j,:]\Big) \\ + \alpha_d \cos\Big(W^d_{\ell,e^{\text{ref}}_\ell}[:,i], W^d_{\ell,e}[:,j]\Big)\,,\end{aligned} \tag{3}$$

where $\alpha_g, \alpha_u, \alpha_d \geq 0$ and $\alpha_g + \alpha_u + \alpha_d = 1$. Combining the three projections accounts for the weights on both the input and output sides of each intermediate channel. The Hungarian algorithm finds the one-to-one assignment maximizing total similarity. Applying this permutation to the rows of the gate and up matrices and the corresponding columns of the down matrix yields aligned projections $\widetilde{W}^g_{\ell,e}$, $\widetilde{W}^u_{\ell,e}$, and $\widetilde{W}^d_{\ell,e}$. These coordinated permutations preserve each expert's function while establishing channel correspondence for parameter fusion.

Shared Subspace Extraction via SVD. To concentrate the selected parameters on a compact set of jointly estimated directions, we construct a shared basis for each projection type $q \in \{g, u, d\}$. Let $\widetilde{W}^q_{\ell,e} \in \mathbb{R}^{n_q \times d_q}$. We concatenate the aligned matrices along their row dimension and compute

$$Z^q_\ell = \text{Concat}_{e \in \mathcal{S}_\ell}\Big(\widetilde{W}^q_{\ell,e}\Big), \qquad Z^q_\ell = U^q_\ell \Sigma^q_\ell (V^q_\ell)^\top. \tag{4}$$

The leading r_q right singular vectors form $B^q_\ell \in \mathbb{R}^{d_q \times r_q}$. This basis minimizes the joint reconstruction error under a shared right-subspace constraint. It captures dominant aggregate parameter directions, which we use as a proxy for reusable structure under the assumption that recurring components dominate weaker expert-specific variations.

Score-Weighted Fusion in the Shared Subspace. To favor experts with stronger cross-domain generality, we use softmax-normalized scores as fusion weights: $\omega_{\ell,e} = \frac{\exp(s_{\ell,e})}{\sum_{e' \in \mathcal{S}_\ell} \exp(s_{\ell,e'})}$. We fuse their coefficients in the shared basis and reconstruct the condensed matrix, equivalently projecting their weighted parameter average:

$$\widehat{W}^q_\ell = \left(\sum_{e \in \mathcal{S}_\ell} \omega_{\ell,e} \widetilde{W}^q_{\ell,e}\right) B^q_\ell (B^q_\ell)^\top. \tag{5}$$

The scores determine expert contributions, while the basis determines the retained directions. Applying this procedure separately to the three projections, each with its own basis, produces a single condensed expert $\widehat{E}^s_\ell = (\widehat{W}^g_\ell, \widehat{W}^u_\ell, \widehat{W}^d_\ell)$. The resulting matrices retain their original dimensions; low-rank projection shapes their initialization.

3.3 Condensed Generalist Experts Injection and Adaptation

To make the condensed components usable by another pretrained model, we introduce a residual expert branch that accommodates different hidden dimensions and layer depths. As shown in the third stage of Figure 2, the branch maps target representations into the source expert space and returns the expert output to the target space, allowing source-derived parameters to participate directly in target computation.

For an injected target layer ℓ, let $\rho(\ell)$ denote its associated source layer and F^t_ℓ its original feed-forward transformation. Given hidden states $H_\ell \in \mathbb{R}^{T \times d_t}$ at the input to this transformation, where T is the token count and d_t the target hidden dimension, we construct a fixed semi-orthogonal matrix

Table 1: Comparison of fine-tuning baselines on language understanding benchmarks. **Train.** denotes the percentage of trainable parameters relative to the total parameters of the target model. For **LLM2llm**, Qwen3.5-122B and Qwen3.5-35B are used as the source MoEs for OLMoE-7B and LLaMA2-7B, respectively. All other baselines operate directly on the target model without using a source model. ESFT† uses an MoE variant constructed from LLaMA2-7B.

Baseline	BoolQ		MMLU		Hellaswag		PIQA		WinoGrande		CaseHold		MedMCQA		Avg.
	Train.	Acc.	Train.	Acc.	Train.	Acc.	Train.	Acc.	Train.	Acc.	Train.	Acc.	Train.	Acc.	
OLMoE-7B															
Full FT	100%	53.52	100%	49.38	100%	85.13	100%	82.37	100%	50.43	100%	82.22	100%	27.28	61.48
LoRA-r4	0.56%	80.52	0.56%	44.68	0.56%	79.39	0.56%	76.28	0.56%	47.83	0.56%	80.22	0.56%	42.79	64.53
LoRA-r8	1.10%	78.72	1.10%	50.03	1.10%	88.03	1.10%	77.75	1.10%	49.41	1.10%	73.96	1.10%	42.86	65.82
PiSSA	0.56%	64.19	0.56%	50.73	0.56%	85.89	0.56%	79.02	0.56%	48.70	0.56%	83.09	0.56%	41.46	64.73
ESFT	2.64%	84.28	2.18%	51.87	2.00%	90.06	1.90%	81.34	1.45%	51.07	2.00%	80.05	1.45%	48.48	69.59
LLM2llm	0.54%	**86.91**	0.54%	**54.15**	0.54%	**92.53**	0.54%	**86.45**	0.54%	**64.01**	0.54%	**84.98**	0.54%	**48.63**	**73.95**
LLaMA2-7B															
Full FT	100%	59.30	100%	33.05	100%	60.73	100%	68.61	100%	49.34	100%	75.62	100%	23.06	52.82
LoRA-r8	0.12%	37.83	0.12%	33.95	0.12%	25.16	0.12%	49.73	0.12%	49.01	0.12%	65.70	0.12%	25.99	41.05
LoRA-r16	0.25%	37.89	0.25%	34.23	0.25%	24.28	0.25%	50.98	0.25%	**50.04**	0.25%	66.25	0.25%	32.44	42.30
PiSSA	0.50%	53.03	0.50%	34.43	0.50%	24.68	0.50%	52.20	0.50%	49.96	0.50%	65.35	0.50%	25.20	43.55
ESFT†	0.63%	80.21	0.63%	38.44	0.63%	69.93	0.63%	69.10	0.63%	49.72	0.63%	74.96	0.63%	36.31	59.81
LLM2llm	0.19%	**81.16**	0.19%	**40.17**	0.19%	**70.97**	0.19%	**70.62**	0.19%	49.96	0.19%	**78.33**	0.19%	**36.65**	**61.12**

$P_\ell \in \mathbb{R}^{d_s \times d_t}$ by QR decomposition of a random Gaussian matrix. It has orthonormal columns when $d_s \geq d_t$ and orthonormal rows otherwise, providing a fixed dimensional interface. The augmented transformation is

$$\begin{aligned} \Delta H_\ell &= \widehat{E}^s_{\rho(\ell)}\big(H_\ell P_\ell^\top\big)\, P_\ell, \\ F_\ell^{\mathrm{aug}}(H_\ell) &= F_\ell^t(H_\ell) + \beta \Delta H_\ell, \end{aligned} \tag{6}$$

where β is a fixed scaling coefficient. The branch processes each token independently of the target's native routing decisions and is therefore compatible with both dense and MoE feed-forward blocks.

Expert selection and condensation are performed once without training. The same condensed experts can initialize adaptation branches for multiple domains and tasks, enabling pretrained target models to inherit and adapt source expert knowledge. During downstream SFT, the target backbone and P_ℓ remain frozen, and only the full gate, up, and down matrices of the injected experts are trained. After extraction, downstream training requires neither the source model nor distillation targets.

4 Experiments

4.1 Baselines and Datasets

Baselines We evaluate LLM2llm on two heterogeneous source–target model pairs. Specifically, we transfer expert knowledge from Qwen3.5-122B-A10B (Qwen Team, 2026) to OLMoE-7B (Muennighoff et al., 2025), and from Qwen3.5-35B-A3B (Qwen Team, 2026) to LLaMA2-7B (Touvron et al., 2023). The two settings span different model scales, backbone architectures, and MoE construction paradigms. OLMoE is pretrained natively as an MoE model, whereas LLaMA2-7B is a dense model. This diversity allows us to evaluate expert transfer across heterogeneous model configurations.

For each target model, we compare LLM2llm against **Full Fine-Tuning (Full FT)**, **LoRA** (Hu et al., 2021), **PiSSA** (Meng et al., 2024), and **ESFT** (Wang et al., 2024). Full FT updates all target-model parameters, while LoRA and PiSSA perform parameter-efficient adaptation through trainable low-rank updates, with PiSSA initializing the low-rank factors from the principal singular components of the pretrained weights. ESFT selectively updates a subset of experts in the MoE model during SFT, making it the most closely related baseline to our expert-based adaptation strategy. For the LLaMA2-7B setting, ESFT is applied to an MoE variant constructed from LLaMA2-7B, whereas LLM2llm and the other baselines adapt the original dense model. We therefore include ESFT as a comparison with an expert-based adaptation approach rather than a baseline using an identical target

Table 2: Comparison of different fine-tuning baselines on dialogue generation benchmarks. All baselines are first supervised fine-tuned on Dolly for 10 epochs and then evaluated on DollyEval, S-NI, UnNI, SelfInst, and Vicuna. ESFT† uses an MoE variant constructed from LLaMA2-7B.

Baseline	Train.	DollyEval	S-NI	UnNI	SelfInst	Vicuna	Avg.
OLMoE-7B							
Full FT	100%	22.58	23.84	26.38	11.48	6.13	18.08
LoRA-r4	0.56%	28.26	30.17	31.51	18.02	17.34	25.06
LoRA-r8	1.10%	29.48	31.61	31.44	18.74	17.29	25.71
PiSSA	0.56%	24.37	33.31	30.97	18.28	16.31	24.65
ESFT	1.50%	29.23	34.32	33.96	18.91	17.38	26.76
LLM2llm	0.54%	**31.75**	**35.02**	**37.00**	**18.95**	**17.74**	**28.09**
LLaMA2-7B							
Full FT	100%	16.27	20.18	21.23	11.80	5.37	14.97
LoRA-r8	0.12%	19.13	25.70	23.30	12.42	5.28	17.17
LoRA-r16	0.25%	19.10	26.20	23.65	13.54	5.32	17.56
PiSSA	0.50%	19.93	27.91	23.58	13.52	5.06	18.00
ESFT†	0.63%	18.60	28.20	23.96	13.42	4.99	17.83
LLM2llm	0.19%	**21.08**	**30.81**	**24.96**	**14.22**	**5.39**	**19.29**

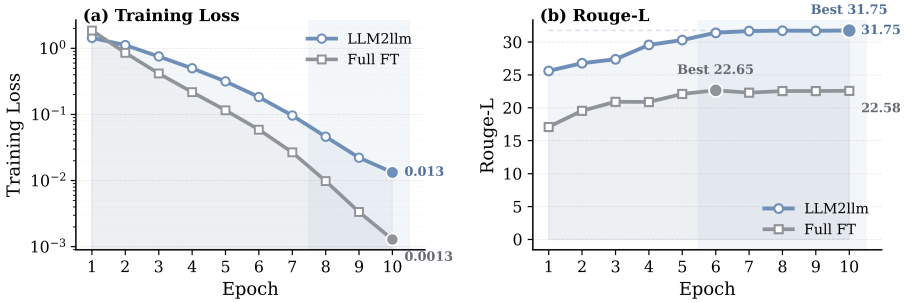


Figure 3: Comparison of the training dynamics between LLM2llm and Full FT on Dolly with OLMoE-7B as the target model, including (a) **training loss** and (b) **Rouge-L** across training epochs.

architecture. For each target model, all baselines use the same downstream datasets, training budget, and training settings.

Datasets We conduct SFT experiments on a broad collection of tasks spanning different task formats and knowledge domains. For language understanding, we use **BoolQ** (Clark et al., 2019), **MMLU** (Hendrycks et al., 2021), **Hellaswag** (Zellers et al., 2019), **PIQA** (Bisk et al., 2020), **WinoGrande** (Sakaguchi et al., 2021), **CaseHold** (Zheng et al., 2021), and **MedMCQA** (Pal et al., 2022). These benchmarks cover reading comprehension, commonsense and physical reasoning, as well as specialized legal and medical knowledge. For dialogue and instruction-following generation, we follow the task suite used in MiniLLM (Gu et al., 2024), including **Dolly**[1], **S-NI** (Wang et al., 2022), **UnNI** (Honovich et al., 2023), **SelfInst** (Wang et al., 2023), and **Vicuna** (Chiang et al., 2023). Together, these datasets provide diverse evaluation scenarios across both discriminative understanding and open-ended generation.

4.2 Main Results

Overall Performance. Tables 1 and 2 report the results on language understanding and dialogue generation benchmarks. LLM2llm achieves the highest *average* score under both source–target settings while updating only a small fraction of the target parameters. On Qwen3.5-122B → OLMoE-7B, LLM2llm reaches average scores of **73.95** and **28.09** on the two task groups, respectively, outperforming ESFT by 4.36 and 1.33 points with only **0.54%** trainable parameters, compared with 2.64% for ESFT. On Qwen3.5-35B → LLaMA2-7B, it achieves average scores of **61.12** and **19.29** with only **0.19%** trainable parameters.

[1]https://github.com/databrickslabs/dolly/tree/master

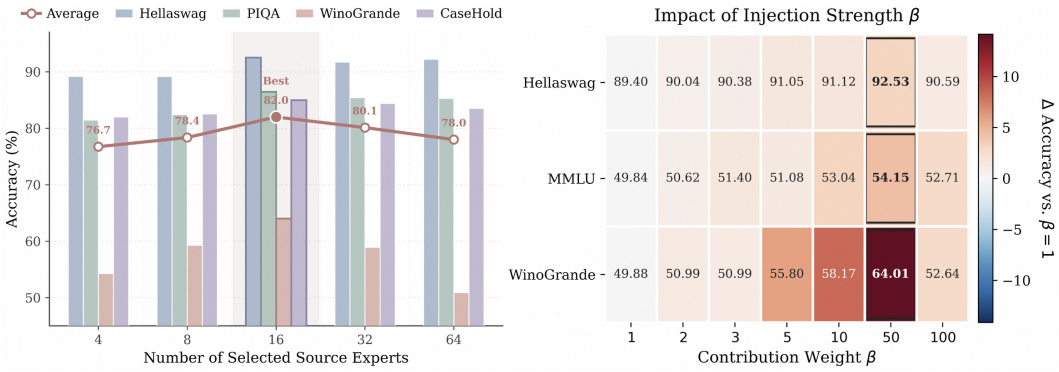


Figure 4: **Left**: Effect of the number of selected source generalist experts m used for expert condensation on downstream performance. **Right**: Effect of the injection strength β on downstream performance using OLMoE-7B as the target model.

Table 3: Ablation of expert selection and condensation on Qwen3.5-122B → OLMoE-7B.

Method	BoolQ	MMLU	PIQA	WinoGrande	CaseHold	Avg.
Random Init.	54.59	36.65	49.46	51.14	70.45	52.46
Random Selection	84.73	52.78	84.11	51.22	83.67	71.30
Direct Fusion	85.14	53.82	85.20	53.20	84.54	72.38
LLM2llm	**86.91**	**54.15**	**86.45**	**64.01**	**84.98**	**75.30**

The advantage over different baselines can be attributed to their distinct adaptation mechanisms. Full FT updates the entire target model and therefore introduces substantially higher optimization capacity, which can be prone to overfitting under limited domain-SFT data. LoRA learns task-specific low-rank updates without incorporating external knowledge, whereas PiSSA improves their initialization using principal components of the target model itself. ESFT similarly adapts only selected native experts, but remains restricted to knowledge already encoded in the target model. In contrast, LLM2llm introduces condensed generalist knowledge extracted from a stronger source MoE and adapts only the transferred experts, providing both a stronger knowledge prior and a substantially smaller adaptation space.

Training Dynamics. Figure 3 compares the training dynamics of LLM2llm and Full FT on Dolly. Full FT converges to a much lower training loss (**0.0013** vs. **0.013**), yet its Rouge-L on DollyEval remains consistently below LLM2llm, which reaches **31.75** and exceeds Full FT by **9.17** points at the final epoch. This discrepancy suggests overfitting in Full FT: the complexity of updating the entire model is excessive relative to the limited domain-SFT data. In contrast, LLM2llm freezes the target backbone and adapts only the source-derived condensed experts, reducing the effective optimization space while introducing transferable knowledge from the source model. This leads to better generalization despite a higher training loss.

4.3 Ablation Studies

Expert Selection and Condensation. We ablate the three key components of LLM2llm under the Qwen3.5-122B → OLMoE-7B setting. **Random Init.** replaces the transferred expert with a randomly initialized expert of the same architecture and size, testing whether the gain comes from source knowledge rather than additional trainable capacity. **Random Selection** randomly selects the same number of source experts while retaining the subsequent condensation and transfer procedure, testing the effectiveness of our cross-domain common-expert selection. **Direct Fusion** directly averages the selected experts instead of applying condensation, testing whether extracting their shared structure is necessary. As shown in Table 3, LLM2llm achieves the highest average score of **75.30**, compared with 52.46, 71.30, and 72.38 for the three variants, respectively. The substantial drop with Random Init. suggests that the gains depend on the transferred initialization beyond additional trainable capacity, while the smaller but consistent drops with Random Selection and Direct Fusion support the contributions of expert selection and condensation. Detailed constructions of these variants are provided in Appendix A.

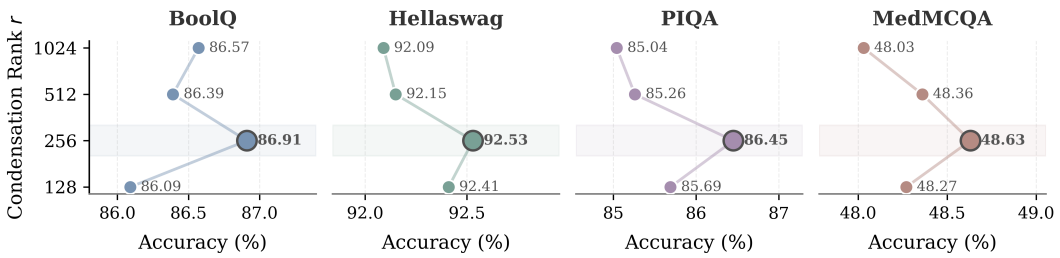


Figure 5: Effect of the retained **rank** r in low-rank expert condensation on downstream performance using OLMoE-7B as the target model.

Table 4: Effect of target-layer injection strategies on downstream performance with OLMoE-7B. Four source layers are selected, and their condensed generalist experts are injected into different sets of four target layers. $*$ denotes the injection strategy adopted in LLM2llm.

Injection Strategy	Hellaswag	PIQA	Winogrande	CaseHold	Avg.
Early Layers	91.07	84.22	60.83	81.41	79.38
Late Layers	80.27	62.73	51.07	59.60	63.42
Random Layers	92.14	83.41	63.14	82.97	80.42
$*$ Uniformly Spaced Layers	**92.53**	**86.45**	**64.01**	**84.98**	**81.99**

We further investigate the number m of selected generalist experts used for condensation in Fig. 4 (left). Performance first improves with m and peaks at $m = 16$, but decreases as more experts are included. Too few experts may provide insufficient coverage of broadly reusable knowledge, whereas a larger m introduces lower-ranked experts with weaker cross-domain generality and more expert-specific or redundant variations. This suggests a trade-off between the breadth and purity of the knowledge aggregated from the source model.

Impact of Key Hyperparameters We examine the condensation rank r and injection strength β on OLMoE-7B. As shown in Fig. 5, $r = 256$ performs best across most benchmarks. Smaller ranks may discard useful shared information, while larger ranks may retain excessive expert-specific variation. Since the expert architecture and trainable parameter count remain unchanged across ranks, these differences reflect initialization quality rather than adaptation capacity, suggesting that selective retention of source parameter directions benefits downstream learning. Figure 4 (right) shows that $\beta = 50$ yields the best overall performance. This suggests that the transferred expert should make a sufficient contribution without overwhelming the original target representations.

Effect of Injection Location We further study where the transferred experts should be injected into the target model. In our default setting, the selected source layers are mapped to uniformly spaced target layers; for example, Table 4 uses four condensed generalist experts extracted from four source layers and injects them into four target layers. We compare this strategy with injecting into early, late, or randomly selected layers. Uniformly spaced injection achieves the best average performance of **81.99**. Since representations evolve across model depth and different layers capture complementary levels of information, concentrating transferred experts only in shallow or deep layers limits their influence to a narrow stage of representation processing. In contrast, uniform injection distributes source-derived knowledge across multiple depths, allowing it to complement the target model throughout the representation hierarchy.

5 Conclusion

We propose LLM2llm, a parameter-efficient framework for transferring expert knowledge from a stronger source MoE to a pretrained target model. LLM2llm identifies cross-domain generalist experts, condenses their shared parameter structure through channel alignment and low-rank fusion, and injects the resulting experts into the target model for downstream adaptation. Across heterogeneous source–target pairs and both understanding and generation tasks, LLM2llm consistently outperforms Full FT and representative PEFT baselines while updating only a small fraction of the target parameters. Overall, our results demonstrate the potential of cross-model expert transfer as an effective way to reuse knowledge from stronger MoE models for efficient downstream adaptation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and debugging portions of our code, provide feedback on mathematical exposition, and check consistency. Additionally, we used these tools to polish the language and improve manuscript clarity. The core research ideas and methodology were developed by the authors, who implemented the method, conducted the experiments, and obtained all reported results. The authors reviewed and verified all AI-assisted work, including code and manuscript revisions, and take full responsibility for the final content of this work, including its text, claims, and artifacts.

ETHICS STATEMENT

This work investigates cross-model expert knowledge transfer using publicly available pretrained models and established benchmark datasets. The study focuses on methodological development and evaluation on existing benchmarks. We have not identified additional ethical concerns specific to the proposed method or the experimental procedures.

REPRODUCIBILITY STATEMENT

Section 3 describes the expert selection, condensation, and injection procedures. Section 4 presents the datasets, baselines, and evaluation metrics. Appendices A and B provide training settings, ablation configurations, model details, and source–target transfer configurations. These descriptions document the method and experimental setup to support reproducibility.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

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, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pp. 2924–2936, 2019.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, 2023.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 4163–4174, 2020.

Dawid Kopiczko, Tijmen Blankevoort, and Yuki Asano. Vera: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, volume 2024, pp. 6815–6835, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Chang Liu, boyu shi, Xu Yang, Qiufeng Wang, and Xin Geng. Inheriting generalizable knowledge from LLMs to diverse vertical tasks. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=m38vyzeoO0.

Chang Liu, Boyu Shi, Xu Yang, and Xin Geng. Xpert: Expert knowledge transfer for effective training of language models. *arXiv preprint arXiv:2605.08842*, 2026b.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024b.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning, 2019. URL https://arxiv.org/abs/1906.10771.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: hints for thin deep nets. arxiv preprint. *arXiv preprint arXiv:1412.6550*, 2014.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Boyu Shi, YiCheng Jiang, Chang Liu, Qiufeng Wang, Xu Yang, and Xin Geng. Chain-based distillation for effective initialization of variable-sized small language models, 2026. URL https://arxiv.org/abs/2605.07783.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, 2022.

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 *ACL*, 2023.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 784–801, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pp. 159–168, 2021.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training. *arXiv preprint arXiv:2406.16554*, 2024. URL https://arxiv.org/abs/2406.16554.

A Experimental Details

Implementation and Training Setup. We implement all methods in PyTorch and conduct the experiments on two NVIDIA H100 GPUs. To ensure a fair comparison, LLM2llm and all competing baselines follow the same training and evaluation protocol under each target model, except for method-specific parameters. For language understanding tasks, namely BoolQ, MMLU, Hellaswag, PIQA, WinoGrande, CaseHold, and MedMCQA, each method is supervised fine-tuned for 3 epochs on the corresponding training set. For dialogue generation, each method is first supervised fine-tuned on Dolly for 10 epochs and subsequently evaluated on DollyEval, S-NI, UnNI, SelfInst, and Vicuna. Unless otherwise specified, we use a learning rate of 2×10^{-5} with cosine decay, a weight decay of 10^{-2}, gradient clipping at 1.0, and a gradient accumulation step of 6. Distributed training is performed with DeepSpeed.

For LLM2llm, expert selection, condensation, and injection are performed in a ***training-free*** preparation stage before downstream SFT. We use approximately 0.05M calibration tokens per domain from corpora including GitHub, arXiv, Wikipedia, C4, and B3G, covering diverse knowledge domains. These texts are used only to collect expert routing statistics through forward passes of the frozen source model. Channel alignment, SVD-based condensation, and injection involve parameter transformations and fixed cross-model projections; none of these preparation steps requires backpropagation or additional model training. Once the condensed expert parameters have been extracted, the source model is no longer required. Downstream SFT updates only the injected experts while keeping the target backbone frozen. This avoids repeated teacher forward passes required by distillation pipelines that query the teacher during student training, and does not require teacher-generated supervision.

Detailed Ablation Settings. To further clarify the role of each component in LLM2llm, we construct three controlled variants under the Qwen3.5-122B $\rightarrow$ OLMoE-7B setting. Across all variants, we keep the target model, downstream training data, injection locations, optimization settings, and training schedule unchanged, and modify only the component being examined.

- **Random Init.** We replace the transferred expert with a randomly initialized expert of the same architecture and parameter size, while keeping the injection locations and training settings unchanged. This variant tests whether the improvement comes from transferred source knowledge rather than merely from additional trainable capacity.
- **Random Selection.** We randomly select the same number of experts from the source MoE, and then apply the same condensation and transfer procedure as LLM2llm. This variant evaluates whether our common-expert selection strategy is more effective than arbitrary source-expert selection.
- **Direct Fusion.** We retain the selected common experts but directly average their parameters instead of applying the proposed condensation procedure. This variant tests whether condensation provides additional benefit beyond simple expert aggregation.

Together, these variants progressively examine the three key factors underlying LLM2llm: whether *source-model knowledge* is useful beyond additional trainable capacity, whether *common-expert selection* identifies more transferable knowledge than random source experts, and whether *expert condensation* provides additional benefit beyond direct parameter fusion.

Table 5: Configurations of the source and target models used in our experiments.

Model	Role	Params.	# Layers	# Experts
Qwen3.5-122B	Source	122B	48	256
OLMoE-7B	Target	7B	16	64
Qwen3.5-35B	Source	35B	40	256
LLaMA2-7B	Target	6.7B	32	–
LLaMA-MoE	For Comparison with ESFT	6.7B	32	16

B Model Details

B.1 Source–Target Model Pairs

We evaluate LLM2llm on two heterogeneous source–target model pairs: **Qwen3.5-122B-A10B** $\rightarrow$ **OLMoE-7B** and **Qwen3.5-35B-A3B** $\rightarrow$ **LLaMA2-7B**. In both settings, a larger source MoE provides pretrained expert knowledge, while the smaller pretrained target model is adapted to downstream tasks through the transferred experts. The two pairs cover different model scales, backbone architectures, and expert configurations, allowing us to evaluate cross-model expert transfer beyond a specific source–target architecture. The detailed architectural configurations of the source and target models are provided in Table 5.

Importantly, the target model in LLM2llm is not restricted to an MoE architecture. Since the transferred expert is injected as an independent residual branch, the method does not rely on native target experts or routing mechanisms and can therefore be applied to both dense and MoE targets.

Qwen3.5-122B-A10B. Qwen3.5-122B-A10B serves as the source model for OLMoE-7B. LLM2llm analyzes its routing behavior on multi-domain calibration data to identify cross-domain generalist experts. The selected experts are then aligned and condensed before being transferred to the target model.

Qwen3.5-35B-A3B. Qwen3.5-35B-A3B is used as the source model for LLaMA2-7B. This setting provides a different source–target scale and architectural configuration. We apply the same expert selection and condensation procedure without updating any parameters of the pretrained source model.

OLMoE-7B. OLMoE-7B is used as the target model in the first setting. Condensed generalist experts extracted from selected source layers are introduced as residual expert branches at selected target layers. During downstream adaptation, *all original parameters of OLMoE-7B are frozen*, and only the injected expert parameters are optimized.

LLaMA2-7B. LLaMA2-7B serves as the dense target model in the second setting. LLM2llm bridges source–target hidden dimensions through fixed projections and injects condensed experts into selected target layers. During downstream adaptation, all original target parameters and projections remain frozen, and only the injected experts are updated. Since ESFT requires native expert modules, we evaluate it on LLaMA-MoE (Zhu et al., 2024), an MoE variant constructed from LLaMA2-7B. LLaMA-MoE is used exclusively for the ESFT baseline; LLM2llm, Full FT, LoRA, and PiSSA use the original dense LLaMA2-7B. Thus, the ESFT comparison involves a different target architecture.

B.2 Source–Target Transfer Configuration

For all main experiments, we uniformly sample four layers from the source MoE and select $m = 16$ cross-domain generalist experts from each sampled layer. The selected experts within each source layer are then aligned and condensed into a single condensed generalist expert, resulting in four transferred experts in total. These experts are injected into four approximately uniformly spaced layers of the target model, preserving their relative depth ordering between the source and target models. For example, OLMoE-7B contains 16 layers, and the four transferred experts are injected

into Layers 1, 5, 9, and 13 (1-based indexing). Unless otherwise specified, this uniformly distributed source–target mapping is used throughout our main experiments.

C Theoretical Motivation for Low-Rank Expert Condensation

To motivate the extraction of reusable expert components, we analyze how joint SVD and parameter fusion can recover recurring structure while suppressing expert-specific variation. We regard this recurring structure as a candidate carrier of knowledge shared across domains. The analysis establishes a parameter-space justification under explicit assumptions; the usefulness of the extracted components for downstream generalization is evaluated empirically.

Joint low-rank approximation. Consider one projection type $q \in \{g, u, d\}$ in a source layer ℓ with m selected experts. We suppress ℓ and q, writing the aligned matrices as $\widetilde{W}_e \in \mathbb{R}^{n\times d}$, $e = 1, \dots, m$, and the retained rank as $r < d$. Following the condensation procedure in Section 3.2, form

$$Z = \begin{bmatrix} \widetilde{W}_1 \\ \vdots \\ \widetilde{W}_m \end{bmatrix} = U\Sigma V^\top, \qquad B = V[:, 1:r], \qquad \Pi = BB^\top. \tag{7}$$

Here $B \in \mathbb{R}^{d\times r}$ has orthonormal columns and Π is its orthogonal projector. The leading right singular vectors solve the joint approximation problem

$$\min_{Q^\top Q = I_r,\, \{C_e\}} \sum_{e=1}^{m} \left\| \widetilde{W}_e - C_e Q^\top \right\|_F^2, \tag{8}$$

where $Q \in \mathbb{R}^{d\times r}$ and $C_e \in \mathbb{R}^{n\times r}$. For fixed Q, the minimizing coefficients are $C_e = \widetilde{W}_e Q$, and the objective becomes

$$\sum_{e=1}^{m} \left\| \widetilde{W}_e (I_d - QQ^\top) \right\|_F^2 = \|Z\|_F^2 - \mathrm{tr}(Q^\top Z^\top Z Q). \tag{9}$$

Maximizing the trace selects the top-r eigenspace of $Z^\top Z$, giving the SVD solution. Thus, the retained subspace optimally represents the selected matrices jointly. This exact property concerns aggregate parameter energy; interpreting it as shared structure requires an additional model of how that energy is distributed.

Shared-structure interpretation. We model the selected, aligned matrices as

$$\widetilde{W}_e = (C_\star + \Delta_e) B_\star^\top + N_e, \qquad B_\star^\top B_\star = I_r, \tag{10}$$

where $W_\star = C_\star B_\star^\top$ is a recurring parameter component, $\Delta_e \in \mathbb{R}^{n\times r}$ describes expert-specific variation within its subspace, and $N_e \in \mathbb{R}^{n\times d}$ is unstructured variation. For each expert, assume

$$\begin{aligned} \mathbb{E}[\Delta_e] &= 0, & \mathbb{E}[N_e] &= 0, \\ \mathbb{E}[\Delta_e^\top N_e] &= 0, & \mathbb{E}[N_e^\top N_e] &= n\sigma^2 I_d, \\ \mathbb{E}[\Delta_e^\top \Delta_e] &= \Xi, \end{aligned} \tag{11}$$

with finite second moments and $\sigma^2 > 0$. The noise is isotropic in its column second moment and may have components both inside and outside the shared subspace. These are assumptions about the matrices after selection and alignment, not properties guaranteed by either operation.

Let $A = Z^\top Z/m$. Expanding Equation 10, the zero-mean and cross-moment conditions eliminate all signal–noise cross terms, yielding

$$\mathbb{E}[A] = B_\star G B_\star^\top + n\sigma^2 I_d, \qquad G = C_\star^\top C_\star + \Xi. \tag{12}$$

If $\delta = \lambda_{\min}(G) > 0$, the top-$r$ eigenspace of $\mathbb{E}[A]$ is exactly $\mathrm{span}(B_\star)$, separated from its orthogonal complement by an eigengap δ. The isotropic term shifts all eigenvalues equally and therefore does not change this subspace.

This population argument also gives a stability condition for the estimated basis. Write $\Pi_\star = B_\star B_\star^\top$ and $\varepsilon = \|A - \mathbb{E}[A]\|_2$, where $\|\cdot\|_2$ denotes the operator norm. If $\varepsilon < \delta/2$, Weyl's inequality and the Davis–Kahan perturbation bound (Yu et al., 2015) give

$$\|\Pi - \Pi_\star\|_2 \le \frac{\varepsilon}{\delta - \varepsilon} \le \frac{2\varepsilon}{\delta}. \tag{13}$$

Indeed, the empirical top-r eigenvalues remain at least $\delta - \varepsilon$ above the population eigenvalue $n\sigma^2$ on the orthogonal complement. Joint SVD can therefore estimate the recurring subspace when the empirical perturbation is small relative to this gap.

Score-weighted consensus. Let $\omega_e = \exp(s_e)/\sum_j \exp(s_j)$ be the fusion weights from Section 3.2 and $\overline{W} = \sum_e \omega_e \widetilde{W}_e$. For the estimated basis B, the fused coefficients satisfy

$$\widehat{C} = \arg\min_C \sum_{e=1}^m \omega_e \|\widetilde{W}_e B - C\|_F^2 = \sum_{e=1}^m \omega_e \widetilde{W}_e B. \tag{14}$$

This follows by expanding the weighted squared distances around their weighted mean. Reconstructing the matrix gives

$$\widehat{W} = \widehat{C} B^\top = \overline{W}\Pi. \tag{15}$$

Thus, the method computes a score-weighted consensus constrained to a jointly estimated subspace. Consistent with the implementation, the basis is obtained from the unweighted stack; the generalist scores determine contributions during fusion.

Suppression of expert-specific variation. To isolate the effect of projection, first consider the ideal estimator $\overline{W}\Pi_\star$ using the true shared subspace. Since $W_\star \Pi_\star = W_\star$, orthogonal decomposition gives the exact identity

$$\|\overline{W} - W_\star\|_F^2 = \|\overline{W}\Pi_\star - W_\star\|_F^2 + \|\overline{W}(I_d - \Pi_\star)\|_F^2. \tag{16}$$

Consequently, projection onto the true shared subspace never increases the parameter estimation error relative to the same weighted average, regardless of how the weights were obtained.

For a quantitative comparison, additionally assume that the residual pairs (Δ_e, N_e) are independent across experts and treat the weights as fixed. With score-dependent weights, the following calculation instead requires these independence and moment assumptions to hold conditionally on the selection and scores. Let $\overline{\Delta} = \sum_e \omega_e \Delta_e$ and $\overline{N} = \sum_e \omega_e N_e$. Then

$$\overline{W}\Pi_\star - W_\star = \overline{\Delta} B_\star^\top + \overline{N}\Pi_\star.$$

Independence and zero means imply $\mathbb{E}[\overline{N}^\top \overline{N}] = n\sigma^2(\sum_e \omega_e^2) I_d$ and $\mathbb{E}\|\overline{\Delta}\|_F^2 = \operatorname{tr}(\Xi)\sum_e \omega_e^2$. Using the vanishing cross moments and $\operatorname{tr}(\Pi_\star) = r$ gives

$$\begin{aligned} \mathbb{E}\|\overline{W} - W_\star\|_F^2 &= \big(\operatorname{tr}(\Xi) + nd\sigma^2\big)\sum_e \omega_e^2, \\ \mathbb{E}\|\overline{W}\Pi_\star - W_\star\|_F^2 &= \big(\operatorname{tr}(\Xi) + nr\sigma^2\big)\sum_e \omega_e^2. \end{aligned} \tag{17}$$

For the same fusion weights, projection reduces the expected error by $n(d-r)\sigma^2 \sum_e \omega_e^2$. Equivalently, it retains a fraction r/d of the expected unstructured-noise energy. This fraction applies to the N_e contribution; in-subspace variation Δ_e is affected by averaging but is preserved by projection.

Finally, the actual estimator uses the learned projector Π. Combining Equation 13 with $\widehat{W} = \overline{W}\Pi$ yields

$$\begin{aligned} \|\widehat{W} - W_\star\|_F &\le \|\overline{W}\Pi_\star - W_\star\|_F + \|\overline{W}\|_F \|\Pi - \Pi_\star\|_2 \\ &\le \|\overline{W}\Pi_\star - W_\star\|_F + \frac{2\varepsilon}{\delta}\|\overline{W}\|_F. \end{aligned} \tag{18}$$

This separates the ideal condensation error from the cost of estimating the subspace. The bound is deterministic for each realization satisfying the gap condition, so it does not require the estimated projector to be independent of the expert matrices.

Discussion. The analysis explains the complementary roles of the condensation steps: channel alignment establishes the correspondence used in parameter fusion, joint SVD estimates recurring parameter directions under the shared-structure model, and weighted averaging forms a consensus within those directions. When recurring components encode features useful across domains and the subspace is estimated accurately, the resulting expert provides a principled initialization for reusing such features in another pretrained model. This link to reusable knowledge is a modeling hypothesis motivated by cross-domain routing and assessed through downstream transfer experiments, rather than a generalization guarantee from SVD alone. The analysis concerns expert initialization; subsequent adaptation updates the full expert matrices without a rank constraint.

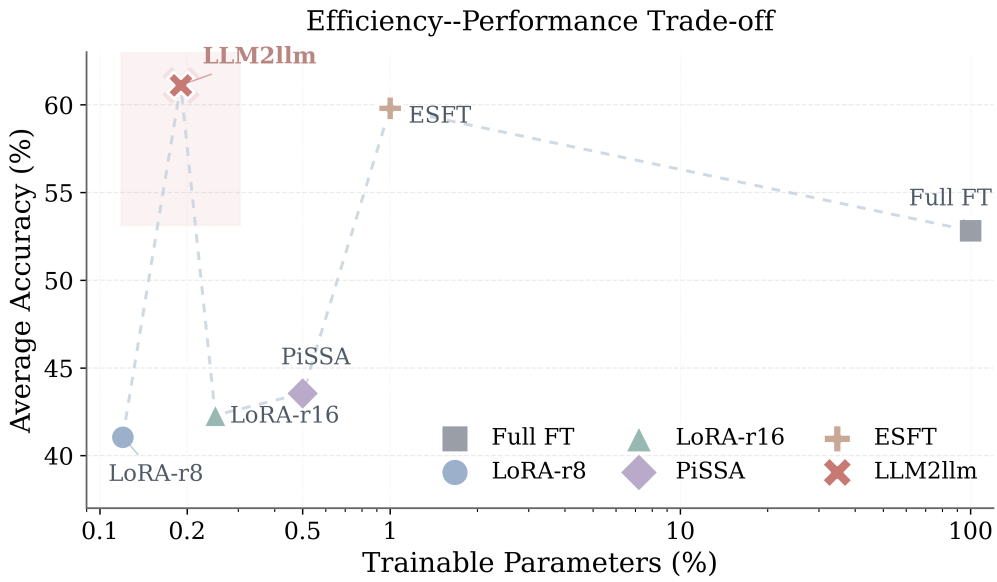


Figure 6: Efficiency–performance trade-off under the Qwen3.5-35B-A3B → LLaMA2-7B setting. The average accuracy across language understanding benchmarks is plotted against the percentage of trainable parameters for different adaptation baselines.

D Efficiency–Performance Trade-off on LLaMA2-7B

We further examine the efficiency–performance trade-off under the Qwen3.5-35B-A3B → LLaMA2-7B setting. Figure 6 compares different adaptation methods in terms of the percentage of trainable parameters and the average accuracy across language understanding benchmarks.

LLM2llm achieves the best overall trade-off, occupying the upper-left region of the figure with substantially higher accuracy while updating only a small fraction of the target-model parameters. Compared with ESFT, LLM2llm achieves higher average accuracy with a considerably smaller trainable-parameter budget. LoRA and PiSSA require similarly small numbers of trainable parameters, but their performance remains substantially lower. Full fine-tuning updates the entire target model while yielding lower accuracy than LLM2llm.

Beyond trainable-parameter efficiency, LLM2llm performs expert selection, condensation, and injection through a *training-free* preparation stage. Selection uses forward-only calibration of the source model; channel alignment, SVD-based condensation, and injection operate through parameter transformations and fixed cross-model projections. These steps require neither backpropagation nor additional model training. Once the condensed expert parameters have been extracted, the source model is no longer needed for injection or downstream adaptation. Subsequent SFT optimizes only the injected experts with the target backbone frozen, without requiring teacher-generated supervision. Compared with distillation pipelines that query a teacher during student training, this design avoids repeated teacher forward passes during downstream SFT.

Together with the corresponding results on OLMoE-7B, these results support the trainable-parameter efficiency of LLM2llm across the evaluated heterogeneous source–target pairs. Its

training-free preparation and source-model-independent downstream adaptation further allow pre-trained expert knowledge to be reused as standalone components, without an additional distillation training stage.

E Additional Comparison with Attention-Only LoRA

Table 6: Comparison with attention-only LoRA on language understanding benchmarks using OLMoE-7B as the target model. LoRA-r4-All applies LoRA to both attention and feed-forward linear layers, whereas LoRA-r150-AttnOnly applies LoRA only to attention projections under a comparable trainable-parameter budget.

Baseline	Train.	BoolQ	MMLU	Hellaswag	PIQA	WinoGrande	CaseHold	MedMCQA	Avg.
LoRA-r4-All	0.56%	80.52	44.68	79.39	76.28	47.83	80.22	42.79	64.53
LoRA-r150-AttnOnly	0.56%	69.08	45.79	81.50	68.55	50.20	70.20	38.06	60.48
LLM2llm	0.54%	**86.91**	**54.15**	**92.53**	**86.45**	**64.01**	**84.98**	**48.63**	**73.95**

Table 7: Comparison with attention-only LoRA on dialogue generation benchmarks using OLMoE-7B as the target model. LoRA-r4-All applies LoRA to both attention and feed-forward linear layers, whereas LoRA-r150-AttnOnly applies LoRA only to attention projections under a comparable trainable-parameter budget. All methods are supervised fine-tuned on Dolly for 10 epochs and evaluated on DollyEval, S-NI, UnNI, SelfInst, and Vicuna.

Baseline	Train.	DollyEval	S-NI	UnNI	SelfInst	Vicuna	Avg.
LoRA-r4-All	0.56%	28.26	30.17	31.51	18.02	17.34	25.06
LoRA-r150-AttnOnly	0.56%	25.76	33.49	31.84	**19.72**	15.05	25.17
LLM2llm	0.54%	**31.75**	**35.02**	**37.00**	18.95	**17.74**	**28.09**

We further compare LLM2llm with an attention-only LoRA variant under a matched trainable-parameter budget on OLMoE-7B. Specifically, **LoRA-r4-All** applies rank-4 LoRA to both attention and feed-forward linear layers, whereas **LoRA-r150-AttnOnly** concentrates a similar number of trainable parameters exclusively in the attention projections, allowing a substantially larger rank.

As shown in Tables 6 and 7, the two LoRA configurations exhibit different behavior across task types. On language understanding, LoRA-r4-All achieves a higher average score than LoRA-r150-AttnOnly (64.53 vs. 60.48), suggesting that restricting adaptation to attention alone limits its ability to adjust the broader representations required by these tasks. In contrast, on dialogue generation, LoRA-r150-AttnOnly slightly improves the average score from 25.06 to 25.17. Under the same parameter budget, concentrating the parameters in attention permits a much larger rank and hence a higher-capacity low-rank update within the attention subspace, which may benefit generation-oriented adaptation. Nevertheless, LLM2llm substantially outperforms both configurations, reaching 73.95 and 28.09 on the two task groups with a comparable trainable-parameter ratio. This suggests that its advantage cannot be explained merely by the allocation or rank of trainable parameters, but is closely related to the transferable knowledge introduced by the source-derived experts.