

Leave It to the Experts: Detecting Knowledge Distillation via MoE Expert Signatures

Pingzhi Li $^{\dagger 1}$, Morris Yu-Chao Huang $^{\dagger 1}$, Zhen Tan 2 , Qingquan Song 3 , Jie Peng 1 , Kai Zou 4 , Yu Cheng 5 , Kaidi Xu 6 , and Tianlong Chen 1

¹ UNC-Chapel Hill ² Arizona State University ³ Individual Contributor ⁴ NetMind.AI ⁵ The Chinese University of Hong Kong ⁶ City University of Hong Kong

Knowledge Distillation (KD) accelerates training of large language models (LLMs) but poses intellectual property protection and LLM diversity risks. Existing KD detection methods based on self-identity or output similarity can be easily evaded through prompt engineering. We present a KD detection framework effective in both white-box and black-box settings by exploiting an overlooked signal: the transfer of MoE "structural habits", especially internal routing patterns. Our approach analyzes how different experts specialize and collaborate across various inputs, creating distinctive fingerprints that persist through the distillation process. To extend beyond the white-box setup and MoE architectures, we further propose Shadow-MoE, a black-box method that constructs proxy MoE representations via auxiliary distillation to compare these patterns between arbitrary model pairs. We establish a comprehensive, reproducible benchmark that offers diverse distilled checkpoints and an extensible framework to facilitate future research. Extensive experiments demonstrate > 94% detection accuracy across various scenarios and strong robustness to prompt-based evasion, outperforming existing baselines while highlighting the structural habits transfer in LLMs.

Code: https://github.com/unites-lab/shadow-moe

1 Introduction

Knowledge Distillation (KD) (Hinton et al., 2015) has emerged as a cornerstone technique for democratizing large language models (LLMs), enabling the transfer of capabilities from computationally expensive and larger teacher models to more efficient and smaller student models. This paradigm has facilitated the training and deployment of powerful AI systems across resource-constrained environments (Gou et al., 2021; Wang & Yoon, 2021; Yang et al., 2025) and accelerated the development of specialized models for domain-specific applications (Xu et al., 2024). However, the widespread adoption of KD has introduced critical challenges to the LLM ecosystem: unauthorized distillation threatens intellectual property rights of model developers (Maini et al., 2021; Li et al., 2025b), while excessive reliance on a few teacher models risks homogenizing the model landscape and stifling innovation (Krishna et al., 2019; Qiu et al., 2025).

Detecting whether a model has undergone knowledge distillation is therefore crucial for both protecting commercial interests and understanding the provenance of AI systems. Existing detection approaches fall into two main categories: *identity-based methods* that probe models' self-identity knowledge (Lee et al., 2025), and *behavior-based methods* that analyze output distribution similarities (Mattern et al.,

[†] Equal Contribution

[™] Correspondence email: {pingzhi, tianlong}@cs.unc.edu

2023). However, these methods exhibit critical limitations. Identity-based approaches can be trivially defeated through prompt engineering or fine-tuning that alters surface-level responses while preserving distilled knowledge. Behavior-based methods struggle with high false positive rates, as models trained on similar data naturally exhibit overlapping behaviors even without distillation (Carlini et al., 2021).

Our work begins with a novel observation: knowledge distillation transfers not merely the functional mapping from inputs to outputs, but also the *structural habits* of the teacher model, *i.e.* the internal computational patterns and decision-making pathways that characterize how the model processes information. Particularly, in Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017; Fedus et al., 2022; Jiang et al., 2024), these structural habits manifest as distinctive expert routing patterns: **expert specialization** of which experts activate for specific input types, and **expert collaboration** of how experts co-activate and cluster, that emerge during training. These routing signatures are deeply embedded in the model's architecture and persist through the distillation process, making them robust indicators of knowledge transfer. This leads to our key research question: Can we leverage the structural signatures inherited through knowledge distillation, particularly the expert routing patterns in MoE models, to reliably detect when distillation has occurred between models?

Recognizing that not all models employ MoE architectures and some only provide API-based text output access, we further introduce Shadow-MoE, a black-box extension that enables KD detection between arbitrary model pairs. Shadow-MoE works by constructing proxy MoE representations of black-box models through further lightweight text-level distillation, *i.e.* training a proxy MoE model to mimic the input-output behavior of target models, thereby exposing accessible routing patterns that preserve the structural habits inherited during knowledge transfer even when direct access to model internals is unavailable.

Our contributions and findings are summarized as follows: (1) We formalize the KD detection task and introduce MoE Expert Signatures (i.e. expert specialization and collaboration), a novel detection method that leverages inherited structural habits in expert routing patterns to identify distillation relationships with accuracy up to 94%. (2) We propose Shadow-MoE, a black-box extension that enables KD detection between arbitrary black-box models by constructing analyzable proxy representations, broadening the applicability beyond MoE architectures and further improving the accuracy to 100%. (3) To our knowledge, we are the first to introduce a benchmark with reproducible experimental protocols and diverse checkpoints, providing the research community with essential infrastructure for advancing distillation detection research.

2 Preliminary

Setting. Let \mathcal{X} denote the input space and \mathcal{Y} the output space. We consider two models: a suspected teacher $f_T: \mathcal{X} \to \Delta(\mathcal{Y})$ and a suspected student $f_S: \mathcal{X} \to \Delta(\mathcal{Y})$, where $\Delta(\mathcal{Y})$ denotes the probability simplex over \mathcal{Y} . We assume black-box query access to both models. Here we define the following Knowledge Distillation Set in Def. 2.1.

Definition 2.1 (Knowledge Distillation Set). The knowledge distillation set $KD(f_T)$ is defined as the set of all possible student model(s) f_S distilled from the teacher model f_T :

$$KD(f_T) := \{ f_S : \exists \mathcal{L}_{KD}, \mathcal{D}_{train}$$
s.t. $f_S = \arg\min_{f} \mathcal{L}_{KD}(f, f_T; \mathcal{D}_{train}) \}$ (2.1)

where \mathcal{L}_{KD} is any knowledge distillation loss (e.g., KL divergence, MSE on logits).

With this, we can define the formulation of the studied knowledge distillation detection below.

2.1 Problem Formulation

We consider a query distribution \mathcal{Q} over $\mathcal{X} \times \mathcal{D}$, where $\mathcal{D} = \{1, \dots, D\}$ indexes semantic domains/tasks (e.g., math, code, medical, etc.)¹. Each sample $(x, d) \sim \mathcal{Q}$ consists of a prompt $x \in \mathcal{X}$ and domain label

¹Domains and tasks are detailed in Section 4.

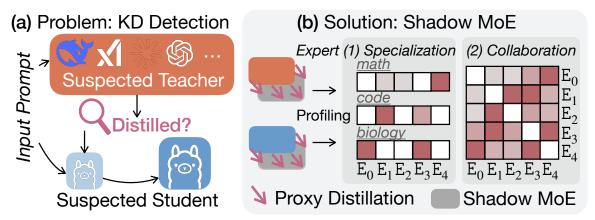


Figure 1: **Overview of our method.** (a) Problem formulation: detecting whether a suspected student model was distilled from a teacher model, which is challenging when only black-box access is available. (b) Our Shadow-MoE solution: we train proxy Shadow-MoE models to mimic both the suspected teacher or student, then analyze their expert routing patterns through two key measurements, *i.e.* expert *specialization*(task-specific activation profiles across different domains) and expert *collaboration* (co-activation patterns between experts). Similar routing patterns between the shadow models provide evidence of a distillation relationship.

 $d \in \mathcal{D}$. We aim to test whether a suspected student f_S has been distilled from a teacher f_T . Formally, the knowledge-distillation detection task is defined as a hypothesis test in Def. 2.2:

Definition 2.2 (Knowledge Distillation Detection). We define the knowledge distillation detection task as a binary hypothesis test:

$$H_1: f_S \in \mathrm{KD}(f_T)$$
 vs. $H_0: f_S \notin \mathrm{KD}(f_T)$,

where $KD(f_T)$ denotes models obtained by distilling from f_T .

Shadow-MoE Construction. Because many models are dense or API-limited, we cannot access their routing directly. We therefore propose to construct the shadow proxies for f_S and f_T that mimic each model's input-output behavior and expose analyzable routing signals as detailed in Def. 2.3.

Definition 2.3 (Shadow-MoE Proxy). A Shadow-MoE proxy $g: \mathcal{X} \to \Delta(\mathcal{Y})$ for model f is a sparse MoE with L layers and E_{ℓ} experts at layer ℓ , trained via:

$$g^* = \arg\min_{g \in \mathcal{G}_{\text{MoE}}} \mathbb{E}_{x \sim \mathcal{Q}_{\mathcal{X}}} \left[\mathcal{L}_{\text{distill}}(g(x), f(x)) \right] + \lambda \Omega(g)$$

The load-balancing regularizer $\Omega(g)$ encourages balanced expert usage across a batch:

$$\Omega(g) = \sum_{\ell=1}^{L} E_{\ell} \sum_{i=1}^{E_{\ell}} \left(\bar{p}_{i}^{(\ell)} - \frac{1}{E_{\ell}} \right)^{2},
\bar{p}_{i}^{(\ell)} = \frac{1}{n} \sum_{m=1}^{n} p_{i}^{(\ell)}(x_{m}),$$
(2.2)

where $p_i^{(\ell)} \in \Delta^{E_\ell}$ is the softmax routing distribution at layer ℓ . This term discourages expert collapse and promotes diverse routing behaviors, following existing works (Fedus et al., 2022; Jiang et al., 2024; DeepSeek-AI, 2025). The load-balancing regularizer encourages each expert to receive a roughly equal fraction of tokens, preventing degenerate proxies where a few experts dominate.

2.2 MoE Expert Specialization and Collaboration

Consider a sparse MoE model (or shadow proxy) g with L layers. At layer $\ell \in [L]$ with E_{ℓ} experts, let the the router outputs gating scores $p^{(\ell)}(x) \in \Delta^{E_{\ell}}$ and selects a top- k_{ℓ} set $\mathcal{K}^{(\ell)}(x) \subseteq \{1, \ldots, E_{\ell}\}$. Define the binary activation for expert i:

$$a_i^{(\ell)}(x) := \mathbb{1}\{i \in \mathcal{K}^{(\ell)}(x)\} \in \{0, 1\}.$$
 (2.3)

We identify two distinct signatures of MoE: Expert Specialization (Li et al., 2023b) and Expert Collaboration (Luo et al., 2025a; Zhang et al., 2025). Below are the definitions of two profiles.

Definition 2.4 (Expert Specialization Profile). For domain $d \in [D]$ with n_d queries and for layer ℓ , define the empirical selection frequency

$$\widehat{S}_{\text{bin},i,d}^{(\ell)} := \frac{1}{n_d} \sum_{m:d_m=d} a_i^{(\ell)}(x_m). \tag{2.4}$$

To compare across domains with possibly varying $k_{\ell}(x)$, we normalize by the expected active expert count

$$\widehat{\kappa}_d^{(\ell)} = \frac{1}{n_d} \sum_{m:d_m=d} k_\ell(x_m), \quad \widehat{\overline{S}}_{i,d}^{(\ell)} = \frac{\widehat{S}_{\mathrm{bin},i,d}^{(\ell)}}{\widehat{\kappa}_d^{(\ell)}},$$

so that each column of $\widehat{\overline{S}}^{(\ell)}$ sums to 1. (If k_{ℓ} is constant, $\widehat{\kappa}_{d}^{(\ell)} = k_{\ell}$.)

Definition 2.5 (Expert Collaboration Matrix). At layer ℓ , the empirical co-activation frequency between experts i and j is

$$\widehat{B}_{i,j}^{(\ell)} := \frac{1}{n} \sum_{m=1}^{n} a_i^{(\ell)}(x_m) a_j^{(\ell)}(x_m), \quad i \neq j,$$
 (2.5)

with $\widehat{B}_{i,i}^{(\ell)} = 0$. To obtain a probability-normalized version, let

$$\widehat{\mathbb{E}}[k_{\ell}(k_{\ell}-1)] = \frac{1}{n} \sum_{m=1}^{n} k_{\ell}(x_{m}) (k_{\ell}(x_{m})-1),$$

$$\widehat{\overline{B}}_{i,j}^{(\ell)} = \frac{\widehat{B}_{i,j}^{(\ell)}}{\widehat{\mathbb{E}}[k_{\ell}(k_{\ell}-1)]},$$
(2.6)

so that $\sum_{i\neq j} \widehat{\overline{B}}_{i,j}^{(\ell)} = 1$ and diagonal remains 0.

The specialization and collaboration profile from Defs. 2.4 and 2.5 are illustrated in Figure 1.

Permutation Invariance. MoE expert labels are arbitrary; two models may differ by permutations yet implement the same routing function. We thus compare specialization and collaboration signatures only via permutation-invariant distances.

Pair Classification Task. Given domain $d \in \{1, ..., 9\}$ (see Section 4.1 for detail domain categories) and a pair of student checkpoints $S_d = \{f_{S,d}^{\text{KD}}, f_{S,d}^{\text{scratch}}\}$. We define f^{scratch} as the model train from scratch without any supervision derived from f_T (e.g., teacher-generated text, hidden states, or reward signals). We cast KD detection as a paired binary classification problem in our experiments in Sections 4.2 and 4.3: The goal is to select the distilled model in each pair. Specifically, each detector produces a scalar score $s(f_T, f_S) \in \mathbb{R}$, where larger values indicate a higher likelihood that f_S is distilled from f_T . For Shadow-MoE, we calculate the average of two signature: specialization d_{spec} and collaboration d_{collab} using the permutation-invariant Wasserstein distance in (3.1) and (3.2). Baselines (e.g. Idiosyncrasies (Sun et al.,

2025)) provide their own monotone scores. We report pairwaise accuracy as Acc = $\frac{1}{9} \sum_{d=1}^{9} \mathbb{1}[\hat{i}_d = \text{KD}]$ and decision margin $m_d = s(f_T, f_{S,d}^{\text{KD}}) - s(f_T, f_{S,d}^{\text{scratch}})$ as metric present in Figures 2 and 3.

3 Methodology

3.1 Proxy Shadow-MoE Training

We consider the problem of detecting whether a suspected student model f_S has been distilled from a teacher model f_T , under the black-box setting. Our key idea is to compare their expert routing signatures, which are invariant to expert index permutations and provide a stable characterization of model behavior. Since many foundation models are not explicitly sparse MoEs, we construct shadow proxies (Def. 2.3) by training sparse MoEs g_T and g_S to mimic f_T and f_S respectively on query-response data. The detection problem then reduces to comparing the specialization and collaboration profiles of g_T and g_S .

3.2 MoE Signature Extraction

For each Shadow-MoE g, we compute two profiles at the last layer ℓ :

- Expert Specialization (Def. 2.4): domain-dependent activation frequencies normalized to probability distributions across experts.
- Expert Collaboration (Def. 2.5): normalized co-activation patterns between expert pairs.

These two metrics capture complementary aspects of expert behavior: *specialization* reflects how domains are partitioned across experts, while *collaboration* reflects how experts jointly contribute within the same domain.

Since expert indices are arbitrary, we measure signature similarity using permutation-invariant Wasserstein distances (Section 2.2). Let $\Pi_{E_{\ell}}$ denote the set of all $E_{\ell} \times E_{\ell}$ permutation matrices. For the ℓ -th MoE layer, we define:

$$d_{\text{spec}}^{(\ell)} = \min_{\Pi \in \Pi_{E_{\ell}}} \frac{1}{D} \sum_{d=1}^{D} W_{1}(\Pi \widehat{\widehat{S}}_{T}^{(\ell)}[:,d], \ \widehat{\widehat{S}}_{S}^{(\ell)}[:,d]), \tag{3.1}$$

$$d_{\text{collab}}^{(\ell)} = \min_{\Pi \in \Pi_{E_{\ell}}} \frac{1}{E_{\ell}} \sum_{i=1}^{E_{\ell}} W_{1} ((\Pi \widehat{\overline{B}}_{T}^{(\ell)} \Pi^{\top})[i,:], \ \widehat{\overline{B}}_{S}^{(\ell)}[i,:]), \tag{3.2}$$

where $W_1(\cdot,\cdot)$ denotes the Wasserstein-1 distance between normalized distributions. In practice, we calculate these distances only at the last MoE layer to obtain overall specialization and collaboration distances, as deeper layer representations often demonstrate more prompt-specific information (Chen et al., 2025; Li et al., 2025a).

3.3 Distillation Detection

We cast our distillation detection as a pair classification task. For each domain d, we receive a candidate pair $S_d = \{f_{S,d}^{\text{KD}}, f_{S,d}^{\text{scratch}}\}$ and select the more likely distilled model by score comparison. We aggregate the two distances by a simple average: score $= -\frac{1}{2}(d_{\text{spec}} + d_{\text{collab}})$, so that higher scores indicate stronger evidence that f_S was distilled from f_T .

In Algorithm 1, we detail a paired KD detection procedure. Given a teacher f_T and a candidate pair $\{f_S^{(1)}, f_S^{(2)}\}$, we query all models on a shared prompt set sampled from \mathcal{Q} . If the teacher or a student is non-MoE or API-limited, we train lightweight Shadow-MoE proxies $(g_T, g_S^{(1)}, g_S^{(2)})$ via Definition 2.3 to expose analyzable routing signals. We then extract expert specialization and collaboration signatures $\Phi(g_T)$ and $\Phi(g_S^{(i)})$, compute the permutation-invariant Wasserstein distances d_{spec} and d_{collab} (Eqs. (3.1), (3.2)), and form a single score $s_i = -\frac{1}{2} (d_{\text{spec}} + d_{\text{collab}})$. The predicted distilled model is $\hat{\imath} = \arg\max_{i \in \{1,2\}} s_i$. Larger scores indicate closer routing similarity to the teacher; we evaluate using pairwise accuracy and decision margins across domains in Section 4.

Algorithm 1 MoE Expert Signature Detection

```
Require: Teacher f_T; student pair \mathcal{S} = \{f_S^{(1)}, f_S^{(2)}\}; query budget n Ensure: Predicted index \hat{\imath} \in \{1, 2\}
  1: Sample \{(x_m, d_m)\}_{m=1}^n \sim \mathcal{Q}
2: if teacher or any student is non-MoE or API-limited then
                                                                                                                                                                    ⊳ shared prompts
              Train proxy g_T to mimic f_T via Def. 2.3
              for each f_S^{(i)} \in \mathcal{S} do
  4:
                    If non-MoE/API-limited, train proxy g_S^{(i)}; else set g_S^{(i)} \leftarrow f_S^{(i)}
  5:
  6:
 8: g_T \leftarrow f_T; g_S^{(i)} \leftarrow f_S^{(i)} \text{ for } i \in \{1, 2\}
9: end if
 10: for i \in \{1, 2\} do
              Extract signatures \Phi(g_T) and \Phi(g_S^{(i)})
Compute d_{\text{spec}}, d_{\text{collab}} via (3.1), (3.2)
Score: s_i \leftarrow -\frac{1}{2}(d_{\text{spec}} + d_{\text{collab}})
 11:
12:
 13:
 14: end for
15: return \hat{\imath} \leftarrow \arg \max_{i \in \{1,2\}} s_i
```

4 Experiments

4.1 Experimental Setup

Calibration Dataset. We construct our calibration dataset by randomly sampling 280 prompts from the allenai/tulu-3-sft-mixture dataset (Lambert et al., 2024), which provides diverse task coverage across multiple domains, including mathematics, coding, and general reasoning. This prompt set serves two purposes in our pipeline: ① Training Shadow-MoE proxies via distillation to mimic the input-output behavior of suspected teacher and student models (Def. 2.3); ② Profiling expert routing patterns to extract specialization and collaboration signatures for detection (Defs. 2.4 and 2.5). The moderate dataset size provides a sweet spot between computational efficiency and sufficient coverage to capture representative routing behaviors across domains.

Model Preparation. We employ DeepSeek-R1 (Guo et al., 2025) as our black-box teacher model, to which we only have access to text outputs without internal information. To construct analyzable proxy representations, we

Table 1: Configuration of the LLMs used in this work.

Model	Top-K	# Shared Experts	# Routed Experts	Model Size
DeepSeek-R1	8	1	256	685B
Moonlight-16B-A3B	6	2	64	16B
OLMoE-1B-7B	8	0	64	7B

train Moonlight-16B-A3B (Liu et al., 2025) as the shadow MoE model using the calibration dataset to mimic the teacher's input-output behavior. For student model evaluation, we use OLMoE-1B-7B (Muennighoff et al., 2024) as the candidate architecture and train it under two conditions, with and without distillation, across 9 domain-specific datasets spanning four categories: Code (TACO, Apps, Code Contests, Codeforces), Math (NuminaMath), Science (Chemistry, Biology, Physics), and Puzzle (Riddle Sense). This yields 18 student checkpoints (9 datasets \times 2 training conditions), enabling comprehensive evaluation of our detection method across diverse domain specializations. Given the two student checkpoints of each dataset, we will apply the baseline methods and our Shadow-MoE to predict which one is distilled from the suspected teacher, as a binary classification task. The configuration of LLMs used in our experiments is presented in Table 1.

Detection Baselines. To validate the effectiveness of our method, we adopt the following baselines for comparison: (1) *Linear model embedding* that extracts response embeddings from candidate models and calculate the cosine similarity between them as distillation score; (2) *BERT embedding* that uses

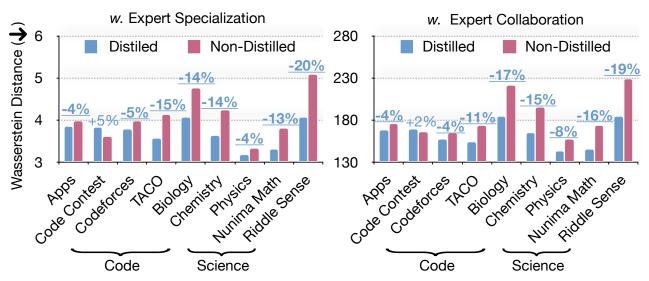


Figure 2: Predicted scores with the black-box teachers and white-box students setting of Shadow-MoE. We show Wasserstein distances between the teacher's Shadow-MoE proxy and student models for both Expert Specialization (left) and Expert Collaboration (right) metrics. Blue bars represent distilled students, while pink bars represent non-distilled students trained from scratch. Percentage differences indicate the relative reduction in distance for distilled models compared to their non-distilled counterparts. Successfully detected tasks (where distilled models show lower distances than non-distilled) are marked with <u>bold underline</u>. Lower distances indicate stronger routing signature similarity, providing evidence of knowledge distillation.

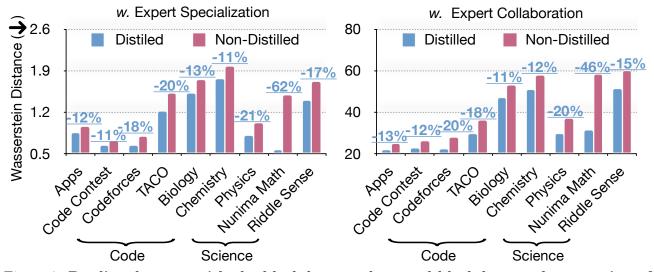


Figure 3: Predicted scores with the black-box teachers and black-box students setting of Shadow-MoE. Same metrics as Figure 2, but with Shadow-MoE proxies constructed for both teacher and student models. Despite the additional proxy approximation for students, the method maintains even stronger detection performance with 100% accuracy between distilled (blue) and non-distilled (pink) models across all tasks.

ModernBERT-base (Warner et al., 2024), a modern BERT-style model, to encode the response from candidate models and calculate the cosine similarity between them as distillation score; (3) *Idiosyncrasies* (Sun et al., 2025) that leverages fine-tuned text embedding models (i.e. LLM2vec) to identify the output patterns across different candidate LLMs by training on held-out teacher-generated responses, i.e. the calibration dataset in our setting; (4) Model self-identity (Lee et al., 2025) that employs jailbreaking techniques, i.e., GPTFuzz (Yu et al., 2024), to probe for identity consistency contradictions, detecting whether a suspected student model inadvertently reveals knowledge of the teacher model's identity through adversarial prompting. The first two baselines rely on surface-level text representations, while the latter two capture behavioral and identity-related signals that may indicate distillation relationships.

4.2 White-box Students, Black-box Teachers

Setting. We first evaluate our Shadow-MoE on a semi-black-box setting, where we have black-box access to the suspected teacher LLMs while white-box access to the suspected student MoE LLMs. Specifically, we construct Shadow-MoE proxies only for the black-box teacher (DeepSeek-R1) using the calibration dataset of 280 prompts, training Moonlight-16B-A3B via text-level distillation for 3

Table 2: Classification accuracies of various methods in *white-box students*, *black-box teachers* setting. We mark the highest accuracy for each task set with **bold**.

Task Set	Linear	BERT	Idiosyncrasies	Self-Identify	Shadow-MoE
Code	50%	50%	50%	0%	75%
Math	100%	100 %	100 %	0%	100%
Science	33%	67%	100 %	0%	100%
Puzzle	0%	100%	100%	0%	100%
Average	46%	54%	88%	0%	94 %

epochs with a learning rate of 5×10^{-6} . For student models, we directly extract routing patterns from the white-box OLMoE-1B-7B checkpoints without requiring proxy construction. Each task set consists of both distilled and non-distilled student models trained on domain-specific data, creating a binary classification problem where we test whether the distilled students align more closely with the teacher than their non-distilled counterparts, and compare baseline methods with ours.

Superior distillation detection performance of Shadow-MoE. Our method achieves an average accuracy of 94% across all task sets, substantially outperforming conventional embedding-based approaches. The performance is particularly strong on Math, Science, and Puzzle tasks, where we achieve 100% accuracy. Notably, the self-identity baseline completely fails, with 0% across all tasks, demonstrating that prompt-based identity probing cannot reliably detect structural knowledge transfer when models are fine-tuned on domain-specific data without identity knowledge.

Consistent separation between distilled and non-distilled models via routing signatures. Figure 2 demonstrates the discriminative effectiveness of our Shadow-MoE approach across diverse domains. Distilled models consistently exhibit lower Wasserstein distances to the teacher's proxy compared to their non-distilled counterparts, with reductions ranging from 4% to 20% for Expert Specialization and 2% to 19% for Expert Collaboration. This pattern holds across all evaluated tasks except for *Code Contest*, where the non-distilled model shows 5% and 2% lower distance, likely due to the code domain inducing similar response structures even without explicit distillation. The complementary nature of the two metrics, with Expert Specialization capturing domain-specific routing preferences and Expert Collaboration revealing inter-expert dependencies, provides echoing evidence for detecting knowledge transfer relationships.

Idiosyncrasies as a competitive baseline. The Idiosyncrasies approach emerges as the strongest one among existing baselines with 88% average accuracy. This method, which trains a text embedding model (i.e., ModernBERT-base) to identify output patterns specific to different LLMs, captures surface-level stylistic signatures that persist through distillation. However, it shows limitations on Code tasks (50% accuracy) where domain-specific syntax and conventions may dominate over model-specific patterns, while routing patterns used in Shadow-MoE provide more consistent signals across diverse domains.

4.3 Black-box Students, Black-box Teachers

Setting. We extend our evaluation to the most challenging pure blackbox setting, where we have only output text access to both the suspected teacher and student models. Unlike Section 4.2 where we could directly extract routing patterns from white-box student MoE models, here we must construct Shadow-MoE proxies for both sides of the detection problem. Specifically, we train Shadow-MoE proxies for both the black-box teacher (DeepSeek-

Table 3: Classification accuracies of various methods in *black-box students*, *black-box teachers* setting. We mark the highest accuracy for each task set with **bold**. The *Linear* baseline, requiring access to hidden states of suspected student models, is not available at this setting.

Task Set	Linear	BERT	Idiosyncrasies	Self-Identify	Shadow-MoE
Code	-	50%	50%	0%	100%
Math	-	100 %	100 %	0%	100%
Science	-	67%	100 %	0%	100 %
Puzzle	-	100%	100%	0%	100%
Average	-	54%	88%	0%	100%

R1) and the black-box student models (OLMoE-1B-7B checkpoints) using the same calibration dataset and training configuration, *i.e.* Moonlight-16B-A3B trained via text-level distillation for 3 epochs with a learning rate of 5×10^{-6} . This introduces an additional layer of approximation for the student models, as we now compare proxy-to-proxy routing signatures rather than proxy-to-actual signatures.

Further improved distillation detection performance of Shadow-MoE in pure black-box setting.

Remarkably, our method achieves perfect detection accuracy of 100% across all task sets in the pure black-box setting, as shown in Table 3, even surpassing its already strong performance in the semi-black-box setting. Figure 3 reveals more pronounced separation between distilled and non-distilled models compared to the white-box student setting, with Wasserstein distance reductions ranging from 11% to 62% for Expert Specialization and 11% to 46% for Expert Collaboration. Notably, even the previously challenging Code Contest task now shows clear separation with 11% and 12% lower distances for the distilled model. This superior performance suggests that Shadow-MoE achieves more precise distillation detection when investing additional computational resources to train proxy models for both teacher and student, likely benefiting from using the same pre-trained model architecture (Moonlight-16B-A3B) as the proxy for both sides.

4.4 Ablation Study and Extended Analysis

Routing Pattern Transferability across Different Distillation and Calibration Tasks.

We investigate whether routing signatures remain discriminative when extracted using different calibration prompt sets than those used during training. We evaluate all 9 training tasks against 28 diverse calibration subsets sampled from various domains within the allenai/tulu-3-sft-mixture dataset. As shown in Figure 4, we measure the relative reduction in Wasserstein distance between distilled and non-distilled models, where more negative values (darker colors) indicate stronger detection signals. Surprisingly, specialized math and code calibration datasets fail to capture significant routing differences even for their corresponding training domains, showing only modest reductions. In contrast, general instruction-following

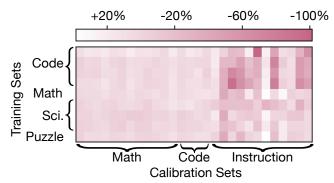


Figure 4: Relative Wasserstein distance reduction for distilled models compared to non-distilled models across different training and calibration set combinations. Darker colors indicate larger reductions (stronger detection signals), with percentages showing how much lower the distilled model's distance is relative to the non-distilled model.

calibration sets consistently achieve strong discriminative power across all task categories, with reductions reaching -60% to -100%. This counterintuitive finding likely suggests that the most informative routing pattern changes induced by distillation occur in the processing of instruction-related tokens rather than domain-specific content.

Routing Efficacy of Different MoE Layers. To validate our choice of using the last MoE layer for signature extraction, we conduct an ablation study comparing routing patterns from different layers in the semiblack-box setting (Section 4.2). We extract expert specialization and collaboration signatures from three positions: {the first, the median, and the last} MoE layer. Table 4 presents the

Table 4: Ablation study on layer selection for routing signature extraction in the white-box students, black-box teachers setting.

Task Set	First Layer	Median Layer	Last Layer (Ours)
Code	50%	75%	75%
Math	100 %	100 %	100 %
Science	33%	67%	100 %
Puzzle	0%	100%	100%
Average	46%	85%	94%

detection accuracy across different task sets. The results demonstrate that deeper layers provide increasingly discriminative routing signatures, with the last layer achieving the highest accuracy of 94%. The first layer shows nearly random discriminative power with 48% accuracy, likely because early routing decisions are more influenced by surface-level token features rather than semantic content. This validates our design choice of using the final layer's routing patterns.

5 Related Works

Mixture-of-Experts (MoE) (Shazeer et al., 2017) has shown promising results for efficiently scaling model capacity without a proportional increase in computational cost. This is typically achieved by replacing dense feed-forward layers with sparse MoE layers, where a routing mechanism directs each input token to a small subset of experts. Switch Transformers (Fedus et al., 2022) simplified MoE routing (i.e., top-1 routing) and demonstrated significant pre-training speedups and scalability up to trillion parameters by reducing communication and computational overheads. Mixtral-8x7B (Jiang et al., 2024) activates only two experts per token per layer but accesses a much larger total parameter count, illustrating that MoE can match the performance of equivalent full-parameter LLMs while utilizing far fewer active parameters. DeepSeek-MoE (Dai et al., 2024; DeepSeek-AI, 2025) refined this architecture with fine-grained expert segmentation and shared experts, aiming for enhanced expert specialization and parameter efficiency. Moreover, expert specialization naturally emerges as the gating network learns to route specific types of inputs to particular experts, reinforcing their proficiency (Dai et al., 2024; Li et al., 2024; Wei et al., 2024). Expert collaboration refers to the co-activation of multiple experts to process certain input tokens, recently enabling reduced communication overhead and efficient expert parallelism through optimized expert placement and routing strategies (Luo et al., 2025b; Zhang et al., 2025). In this work, we leverage expert specialization and collaboration as the underlying functional similarity inherited through distillation for detecting knowledge distillation.

Knowledge Distillation (KD) (Hinton et al., 2015) has been a widely adopted model compression technique where a smaller "student" model is trained to replicate the behavior and inherit the capabilities of a larger, more powerful "teacher" model, to produce efficient yet powerful models (Hsieh et al., 2023; Ma et al., 2021; 2022; Sanh et al., 2019). In the context of LLMs, KD is usually performed at three levels of granularity, including: (1) layer hidden states-level KD for aligning the student's intermediate hidden state representations with those of the teacher (Chang et al., 2022; Liang et al., 2023; Lin et al., 2023), (2) logits-level KD for matching the teacher's final output probability distributions over tokens (Anshumann et al., 2025; Li et al., 2024; Yang et al., 2024), and (3) output text-level KD for replicating the teacher's generated text (Bercovich et al., 2025; Muennighoff et al., 2025; Savani et al., 2025). In this work, we focus on the most widely adopted **output text-level** KD as it is flexible to different student-teacher vocabularies or even black-box models with only API access, and produces minimal computing overhead (Guo et al., 2025; Muennighoff et al., 2025). Recently, KD has gathered significant attention due to the rich semantic information in LLM reasoning traces, which has proven highly effective for transferring complex problem-solving abilities (Guo et al., 2025; Bercovich et al., 2025; Muennighoff et al., 2025; Savani et al., 2025). However, it raises critical concerns about intellectual property protection and model homogenization (Savani et al., 2025). Therefore, there is a growing need to quantify the extent of distillation and develop effective methods to detect if a model has been distilled from another (Lee et al., 2025).

Tracing LLMs to training data coalesce around memorization/extraction, contamination/deduplication, and training-data attribution. Black-box extraction attacks show that individual training sequences can be recovered from deployed LMs and that vulnerability scales with model size (Carlini et al., 2021). Follow-up measurement work quantifies how memorization grows with model capacity, duplication, and prompt context length (Carlini et al., 2023). To curb regurgitation and evaluation inflation, deduplication reduces verbatim emission and train—test overlap (Lee et al., 2022) and directly mitigates extraction risk (Kandpal et al., 2022). Beyond aggregate leakage, Akyürek et al. (2022) formalize fact tracing, retrieving "proponent" training examples for generated assertions, and find that popular gradient- and embedding-based methods still lag strong IR baselines. For scalable per-example attribution, gradient-tracing via TracIn (Pruthi et al., 2020) and randomly projected after-kernel scoring via TRAK (Park et al., 2023) estimate pointwise influence and scale to modern LLMs and CLIP-style VLMs. Collectively, these works motivate provenance-aware analyses when linking behaviors to pretraining corpora; in contrast, our paper pivots to model-internal signals, which use MoE routing patterns as fingerprints to detect knowledge distillation relationships.

6 Conclusion

We introduce a practical framework for detecting knowledge distillation that leverages Mixture-of-Experts routing signatures as structural fingerprints of model behavior. Our approach rests on two key ideas: (i) distillation transfers not only surface behavior but also structural habits in computation, and (ii) these habits can be exposed and compared through lightweight Shadow-MoE proxies even in black-box settings. Concretely, we defined two complementary routing profiles, *i.e. expert specialization* and expert collaboration, and compared them via permutation-invariant Wasserstein distances for distillation detection. Across semi-black-box (*i.e.* black-box teachers and white-box MoE students) and pure black-box (*i.e.* black-box teachers and black-box students) settings, our method consistently outperforms embedding- and identity-based baselines, achieving high accuracy across diverse domains. We release the benchmark with distilled and non-distilled checkpoints to facilitate future study. We see this work as a step toward structure-aware alignment and defenses (*e.g.*, structural watermarks, routing randomization).

Limitations

Our results suggest that structural fingerprints provide a promising path toward provenance analysis for modern LLMs, complementing existing approaches based on identity prompts, text embeddings, or membership signals. Looking ahead, we see three natural directions: **Beyond MoE and richer structure** by extending signature to dense model and incorporate additional structure cues (e.g. attention head usage). **Alternative distillation channels** for detecting reward-model-mediated or RL-based distillation. **Stronger guarantees and defenses** by exploring defensive mechanisms (e.g. structural watermarks or routing randomization) to deter unauthorized distillation.

Acknowledgments

Pingzhi Li, Morris Yu-Chao Huang, and Tianlong Chen are partially supported by Amazon Research Award and Cisco Faculty Award.

References

Akyürek, E., Bolukbasi, T., Liu, F., Xiong, B., Tenney, I., Andreas, J., and Guu, K. Towards tracing knowledge in language models back to the training data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2429–2446. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.180. URL https://aclanthology.org/2022.findings-emnlp.180/.

Anshumann, Zaidi, M. A., Kedia, A., Ahn, J., Kwon, T., Lee, K., Lee, H., and Lee, J. Sparse logit sampling: Accelerating knowledge distillation in llms, 2025. URL https://arxiv.org/abs/2503.16870.

Bercovich, A., Levy, I., Golan, I., Dabbah, M., El-Yaniv, R., Puny, O., Galil, I., Moshe, Z., Ronen, T., Nabwani, N., et al. Llama-nemotron: Efficient reasoning models. arXiv preprint arXiv:2505.00949, 2025. 10

- Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In 30th USENIX security symposium (USENIX Security 21), pp. 2633–2650, 2021. 2, 11
- Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK. 11
- Chang, H.-J., Yang, S.-w., and Lee, H.-y. Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert. In *ICASSP 2022-2022 IEEE International Conference on Acoustics*, Speech and Signal Processing (ICASSP), pp. 7087–7091. IEEE, 2022. 10
- Chen, R., Zhang, Z., Hong, J., Kundu, S., and Wang, Z. Seal: Steerable reasoning calibration of large language models for free, 2025. URL https://arxiv.org/abs/2504.07986. 5
- Dai, D., Deng, C., Zhao, C., Xu, R. X., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., Xie, Z., Li, Y. K., Huang, P., Luo, F., Ruan, C., Sui, Z., and Liang, W. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.
- DeepSeek-AI. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437. 3, 10
- Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961. 2, 3, 10
- Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021. 1
- Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025. 6, 10
- Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., Burns, C., Puranik, S., He, H., Song, D., and Steinhardt, J. Measuring coding challenge competence with apps. *NeurIPS*, 2021. 16
- Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531. 1, 10
- Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. arXiv preprint arXiv:2305.02301, 2023. 10
- Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088. 2, 3, 10
- Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11220–11234. PMLR, 2022. URL https://proceedings.mlr.press/v162/kandpal22a/kandpal22a.pdf. 11
- Krishna, K., Tomar, G. S., Parikh, A. P., Papernot, N., and Iyyer, M. Thieves on sesame street! model extraction of bert-based apis. arXiv preprint arXiv:1910.12366, 2019. 1
- Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tülu 3: Pushing frontiers in open language model post-training. 2024. 6, 16

- Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/. 11
- Lee, S., Zhou, J., Ao, C., Li, K., Du, X., He, S., Wu, H., Liu, T., Liu, J., Alinejad-Rokny, H., Yang, M., Liang, Y., Wen, Z., and Ni, S. Quantification of large language model distillation, 2025. URL https://arxiv.org/abs/2501.12619. 1, 8, 10
- Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. Camel: Communicative agents for "mind" exploration of large scale language model society, 2023a. 16
- LI, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S. C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., and Polu, S. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024. 16
- Li, P., Zhang, Z., Yadav, P., Sung, Y.-L., Cheng, Y., Bansal, M., and Chen, T. Merge, then compress: Demystify efficient smoe with hints from its routing policy. arXiv preprint arXiv:2310.01334, 2023b. 4
- Li, P., Zhang, Z., Yadav, P., Sung, Y.-L., Cheng, Y., Bansal, M., and Chen, T. Merge, then compress: Demystify efficient smoe with hints from its routing policy, 2024. URL https://arxiv.org/abs/2310.01334. 10
- Li, P., Jin, X., Tan, Z., Cheng, Y., and Chen, T. Quantmoe-bench: Examining post-training quantization for mixture-of-experts, 2025a. URL https://arxiv.org/abs/2406.08155. 5
- Li, P., Tan, Z., Qu, H., Liu, H., and Chen, T. Doge: Defensive output generation for llm protection against knowledge distillation. arXiv preprint arXiv:2505.19504, 2025b. 1
- Li, R., Fu, J., Zhang, B.-W., Huang, T., Sun, Z., Lyu, C., Liu, G., Jin, Z., and Li, G. Taco: Topics in algorithmic code generation dataset. arXiv preprint arXiv:2312.14852, 2023c. 16
- Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D., Sutherland Robson, E., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. arXiv preprint arXiv:2203.07814, 2022. 16
- Liang, C., Zuo, S., Zhang, Q., He, P., Chen, W., and Zhao, T. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pp. 20852–20867. PMLR, 2023. 10
- Lin, B. Y., Wu, Z., Yang, Y., Lee, D.-H., and Ren, X. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. 2021. 16
- Lin, Y.-J., Chen, K.-Y., and Kao, H.-Y. Lad: Layer-wise adaptive distillation for bert model compression. Sensors, 23(3):1483, 2023. 10
- Liu, J., Su, J., Yao, X., Jiang, Z., Lai, G., Du, Y., Qin, Y., Xu, W., Lu, E., Yan, J., Chen, Y., Zheng, H., Liu, Y., Liu, S., Yin, B., He, W., Zhu, H., Wang, Y., Wang, J., Dong, M., Zhang, Z., Kang, Y., Zhang, H., Xu, X., Zhang, Y., Wu, Y., Zhou, X., and Yang, Z. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982.
- Luo, S., Li, P., Peng, J., Wang, H., Cheng, Y., Chen, T., et al. Occult: Optimizing collaborative communication across experts for accelerated parallel moe training and inference. arXiv preprint arXiv:2505.13345, 2025a. 4

- Luo, S., Li, P., Peng, J., Zhao, Y., Cao, Y., Cheng, Y., and Chen, T. Occult: Optimizing collaborative communications across experts for accelerated parallel moe training and inference. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=vh2Dt4sT67. 10
- Ma, H., Chen, T., Hu, T.-K., You, C., Xie, X., and Wang, Z. Undistillable: Making a nasty teacher that cannot teach students, 2021. URL https://arxiv.org/abs/2105.07381. 10
- Ma, H., Huang, Y., Chen, T., Tang, H., You, C., Wang, Z., and Xie, X. Stingy teacher: Sparse logits suffice to fail knowledge distillation, 2022. URL https://openreview.net/forum?id=ae7BJIOxkxH. 10
- Maini, P., Yaghini, M., and Papernot, N. Dataset inference: Ownership resolution in machine learning. arXiv preprint arXiv:2104.10706, 2021. 1
- Mattern, J., Mireshghallah, F., Jin, Z., Schölkopf, B., Sachan, M., and Berg-Kirkpatrick, T. Membership inference attacks against language models via neighbourhood comparison. arXiv preprint arXiv:2305.18462, 2023. 1
- Muennighoff, N., Soldaini, L., Groeneveld, D., Lo, K., Morrison, J., Min, S., Shi, W., Walsh, P., Tafjord, O., Lambert, N., Gu, Y., Arora, S., Bhagia, A., Schwenk, D., Wadden, D., Wettig, A., Hui, B., Dettmers, T., Kiela, D., Farhadi, A., Smith, N. A., Koh, P. W., Singh, A., and Hajishirzi, H. Olmoe: Open mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2409.02060. 6
- Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling. arXiv preprint arXiv:2501.19393, 2025.
- Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c/park23c.pdf. 11
- Penedo, G., Lozhkov, A., Kydlíček, H., Allal, L. B., Beeching, E., Lajarín, A. P., Gallouédec, Q., Habib, N., Tunstall, L., and von Werra, L. Codeforces. https://huggingface.co/datasets/open-r1/codeforces, 2025. 16
- Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf. 11
- Qiu, S., Guo, S., Song, Z.-Y., Sun, Y., Cai, Z., Wei, J., Luo, T., Yin, Y., Zhang, H., Hu, Y., et al. Phybench: Holistic evaluation of physical perception and reasoning in large language models. arXiv preprint arXiv:2504.16074, 2025. 1
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019. 10
- Savani, Y., Trockman, A., Feng, Z., Schwarzschild, A., Robey, A., Finzi, M., and Kolter, J. Z. Antidistillation sampling, 2025. URL https://arxiv.org/abs/2504.13146. 10
- Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538. 2, 10
- Sun, M., Yin, Y., Xu, Z., Kolter, J. Z., and Liu, Z. Idiosyncrasies in large language models, 2025. URL https://arxiv.org/abs/2502.12150. 4, 8
- Wang, L. and Yoon, K.-J. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6): 3048–3068, 2021. 1

- Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., and Poli, I. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL https://arxiv.org/abs/2412.13663.
- Wei, T., Zhu, B., Zhao, L., Cheng, C., Li, B., Lü, W., Cheng, P., Zhang, J., Zhang, X., Zeng, L., Wang, X., Ma, Y., Hu, R., Yan, S., Fang, H., and Zhou, Y. Skywork-moe: A deep dive into training techniques for mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2406.06563.
- Xu, X., Li, M., Tao, C., Shen, T., Cheng, R., Li, J., Xu, C., Tao, D., and Zhou, T. A survey on knowledge distillation of large language models. arXiv preprint arXiv:2402.13116, 2024. 1
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. 1
- Yang, C., Zhu, Y., Lu, W., Wang, Y., Chen, Q., Gao, C., Yan, B., and Chen, Y. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 2024. 10
- Yu, J., Lin, X., Yu, Z., and Xing, X. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts, 2024. URL https://arxiv.org/abs/2309.10253. 8
- Zhang, M., Li, P., Peng, J., Qiu, M., and Chen, T. Advancing moe efficiency: A collaboration-constrained routing (c2r) strategy for better expert parallelism design, 2025. URL https://arxiv.org/abs/2504.01337. 4, 10

Appendix

A	Experiment Details	16
В	Dataset Details	16
	Details of Distance Metrics for Routing Pattern Comparison C.1 Expert Specialization Distance C.2 Expert Collaboration Distance C.3 Aggregate Detection Score	17

A Experiment Details

Experiments were conducted on NVIDIA A100 and B200 GPU servers. For all training runs, we use the AdamW optimizer with a weight decay of 0.01 and a warm-up ratio of 0.1. For all MoE models, we apply a load-balancing loss with a coefficient of 0.001. We apply all distillation experiments for 3 epochs with the learning rate of 5×10^{-6} and the batch size of 256. We apply cosine learning rate schedulers.

B Dataset Details

We list the datasets we used in this work and their license here:

- Tulu3 (Lambert et al., 2024) with ODC-BY-1.0 license.
- TACO (Li et al., 2023c) with Apache 2.0 license.
- Apps (Hendrycks et al., 2021) with MIT license.
- Code Contests (Li et al., 2022) with CC-by-4.0 license
- Codeforces (Penedo et al., 2025) with CC-by-4.0 license
- NuminaMath (LI et al., 2024) with Apache 2.0 license
- Chemistry (Li et al., 2023a) with CC-by-NC-4.0 license
- Biology (Li et al., 2023a) with CC-by-NC-4.0 license
- Physics (Li et al., 2023a) with CC-by-NC-4.0 license
- Riddle Sense (Lin et al., 2021)

C Details of Distance Metrics for Routing Pattern Comparison

In this section, we provide detailed mathematical formulations and computational procedures for the Wasserstein distance metrics used to compare expert routing patterns between models.

C.1 Expert Specialization Distance

Given two models (teacher g_T and student g_S) with expert specialization profiles $\widetilde{S}_T^{(\ell)}$ and $\widetilde{S}_S^{(\ell)}$ at layer ℓ (Definition 2.4), we compute the permutation-invariant Wasserstein distance to measure their similarity. For a specific domain $d \in \mathcal{D}$, the normalized specialization profiles $\widetilde{S}_T^{(\ell)}[:,d] \in \Delta^{E_\ell}$ and $\widetilde{S}_S^{(\ell)}[:,d] \in \Delta^{E_\ell}$ represent probability distributions over E_ℓ experts, where each column sums to 1 as specified in Definition 2.4.

The Wasserstein-1 distance between two discrete distributions on expert indices is computed as:

$$W_1(\widetilde{S}_T^{(\ell)}[:,d], \widetilde{S}_S^{(\ell)}[:,d]) = \inf_{\gamma \in \Gamma} \sum_{i=1}^{E_{\ell}} \sum_{j=1}^{E_{\ell}} |i-j| \cdot \gamma_{i,j}$$
 (C.1)

where $\Gamma = \Gamma(\widetilde{S}_T^{(\ell)}[:,d],\widetilde{S}_S^{(\ell)}[:,d])$ is the set of all joint distributions with marginals $\widetilde{S}_T^{(\ell)}[:,d]$ and $\widetilde{S}_S^{(\ell)}[:,d]$. In practice, we use the optimal transport formulation implemented in scipy.stats.wasserstein_distance, which takes expert positions $\mathbf{pos} = [0,1,\ldots,E_\ell-1]$ as the ground metric.

Since expert indices are arbitrary permutations of the same underlying functionality, we compute the optimal permutation-invariant distance as defined in Equation (3.1):

$$d_{\text{spec}}^{(\ell)} = \min_{\Pi \in \Pi_{E_{\ell}}} \frac{1}{D} \sum_{d=1}^{D} W_1 \left(\Pi \widetilde{S}_T^{(\ell)}[:, d], \widetilde{S}_S^{(\ell)}[:, d] \right)$$
 (C.2)

where $\Pi_{E_{\ell}}$ denotes the set of all $E_{\ell} \times E_{\ell}$ permutation matrices. The optimization over permutations is solved using the Hungarian algorithm, which finds the optimal assignment in $\mathcal{O}(E_{\ell}^3)$ time by minimizing the total cost across all domains.

C.2 Expert Collaboration Distance

For expert collaboration patterns $\widetilde{B}_T^{(\ell)}$ and $\widetilde{B}_S^{(\ell)}$ (Definition 2.5), we measure similarity through permutation-invariant Wasserstein distance. The normalized collaboration matrix $\widetilde{B}^{(\ell)} \in [0,1]^{E_\ell \times E_\ell}$ captures pairwise expert co-activation frequencies, where $\sum_{i \neq j} \widetilde{B}_{i,j}^{(\ell)} = 1$ and the diagonal is zero. Each row $\widetilde{B}^{(\ell)}[i,:]$ represents the probability distribution of expert i collaborating with other experts.

To compute the Wasserstein distance between collaboration patterns, we treat the collaboration matrix as a collection of probability distributions. For computational efficiency, we represent the collaboration patterns as sparse dictionaries mapping expert pairs to co-occurrence probabilities:

$$\mathcal{B}_T = \{ (i, j) \mapsto \widetilde{B}_{T, i, j}^{(\ell)} : i \neq j, i, j \in [E_\ell] \}$$
 (C.3)

For a specific expert i, we extract the row vector $\widetilde{B}_{T}^{(\ell)}[i,:]$ and compute its Wasserstein distance to the corresponding row in the student model. The computation proceeds as follows. First, identify all expert pairs that have non-zero collaboration in either model:

$$\mathcal{P}_i = \{ j : \widetilde{B}_{T,i,j}^{(\ell)} > 0 \text{ or } \widetilde{B}_{S,i,j}^{(\ell)} > 0, j \neq i \}$$
 (C.4)

Second, construct aligned probability vectors by extracting collaboration probabilities for all pairs in \mathcal{P}_i , with missing entries defaulting to zero, then normalize to ensure valid probability distributions:

$$\mathbf{p}_{T,i} = [\widetilde{B}_{T,i,j_1}^{(\ell)}, \dots, \widetilde{B}_{T,i,j_{|\mathcal{P}_i|}}^{(\ell)}]^T, \quad \widehat{\mathbf{p}}_{T,i} = \frac{\mathbf{p}_{T,i}}{\|\mathbf{p}_{T,i}\|_1}$$
(C.5)

with analogous construction for $\hat{\mathbf{p}}_{S,i}$. Third, compute the Wasserstein distance using position indices:

$$W_1(\widetilde{B}_T^{(\ell)}[i,:], \widetilde{B}_S^{(\ell)}[i,:]) = \text{wasserstein_distance}([0,\ldots,|\mathcal{P}_i|-1], [0,\ldots,|\mathcal{P}_i|-1], \widehat{\mathbf{p}}_{T,i}, \widehat{\mathbf{p}}_{S,i})$$
(C.6)

Following Equation (3.2), the permutation-invariant collaboration distance averages over all expert rows after applying the optimal permutation:

$$d_{\text{collab}}^{(\ell)} = \min_{\Pi \in \Pi_{E_{\ell}}} \frac{1}{E_{\ell}} \sum_{i=1}^{E_{\ell}} W_1 \left((\Pi \widetilde{B}_T^{(\ell)} \Pi^T)[i,:], \widetilde{B}_S^{(\ell)}[i,:] \right)$$
 (C.7)

where $\Pi \widetilde{B}_T^{(\ell)} \Pi^T$ applies the same permutation to both rows and columns of the collaboration matrix to maintain consistency in expert indexing.

C.3 Aggregate Detection Score

The final detection score combines both specialization and collaboration distances from the last MoE layer $\ell = L$:

$$s(f_T, f_S) = -\frac{1}{2} \left(d_{\text{spec}}^{(L)} + d_{\text{collab}}^{(L)} \right) \tag{C.8}$$

where higher scores indicate stronger routing similarity and thus higher likelihood of a distillation relationship. In the pairwise classification task (Section 2.2), given a candidate pair $\{f_S^{\text{KD}}, f_S^{\text{scratch}}\}$, we select the model with the higher score as the distilled candidate:

$$\widehat{\imath} = \underset{i \in \{\text{KD,scratch}\}}{\arg \max} s(f_T, f_S^{(i)}) \tag{C.9}$$

The computational complexity is $\mathcal{O}(E_L^3 \cdot D)$ for specialization (Hungarian algorithm over D domains) and $\mathcal{O}(E_L^4)$ for collaboration (permutation matching over E_L expert rows), yielding a total complexity of $\mathcal{O}(E_L^4 + E_L^3 D)$ per model pair. For our experiments with $E_L = 64$ experts and D = 9 domains, the computation completes within minutes on standard hardware.