MTA: Multi-Granular Trajectory Alignment for Large Language Model Distillation

Pham Khanh Chi[1]*, **Quoc Phong Dao**[1]*, **Thuat Nguyen**[1], **Linh Ngo Van**[1,†], **Trung Le**[2], **Thanh Nguyen**[3]

[1]Hanoi University of Science and Technology,
[2]Monash University, [3]University of Oregon

Abstract

Knowledge distillation is a key technique for compressing large language models (LLMs), but most existing methods align representations at fixed layers or token-level outputs, ignoring how representations evolve across depth. As a result, the student is only weakly guided to capture the teacher's internal relational structure during distillation, which limits knowledge transfer. To address this limitation, we propose **Multi-Granular Trajectory Alignment (MTA)**, a framework that aligns teacher and student representations along their layer-wise transformation trajectory. MTA adopts a layer-adaptive strategy: lower layers are aligned at the word level to preserve lexical information, while higher layers operate on phrase-level spans (e.g., noun and verb phrases) to capture compositional semantics. We instantiate this idea through a Dynamic Structural Alignment loss that matches the relative geometry among semantic units within each layer. This design is motivated by empirical findings that Transformer representations become increasingly abstract with depth, and is also consistent with linguistic views in which higher-level meaning emerges through the composition of lower-level lexical units. We further incorporate a Hidden Representation Alignment loss to directly align selected teacher–student layers. Experiments show that MTA consistently outperforms state-of-the-art baselines on standard benchmarks, with ablations confirming the contribution of each component.

1 Introduction

While Large Language Models (LLMs) excel in NLP tasks, their computational cost necessitates compression techniques (Xu et al., 2024). To address this problem, Knowledge Distillation (KD) (Hinton et al., 2015) has emerged as a pivotal model compression paradigm, where a smaller "student" model is trained to mimic the behavior of a larger, more capable "teacher" model. Conventional KD approaches for LLMs often focus on minimizing the divergence between the output probability distributions of the teacher and the student (Agarwal et al., 2024; Ko et al., 2024; Le et al.; Anshumann et al., 2025; Nguyen et al., 2026). While effective, these methods overlook the structural knowledge embedded within the intermediate representations. To address this, feature-based distillation methods have been proposed to align internal hidden states or attention maps of the student with those of the teacher (Jiao et al., 2019; Wang et al., 2020; Gong et al., 2025; Vu et al., 2026a). However, most existing approaches adopt a static and uniform alignment strategy across layers, typically operating at the token level. By treating tokens largely in isolation and applying the same objective uniformly, these methods fail to account for differences in token importance and semantic roles across layers, which can harm generalization.

Crucially, this uniform, token-level treatment is misaligned with the hierarchical nature of human language. Linguistic research has long established that natural language is organized hierarchically, composing discrete lexical units into nested phrases and higher-level semantic structures (Chomsky, 1965; Crain and Nakayama, 1987; Hale et al., 2018). Prior analyses of BERT have identified a functional hierarchy in which lower layers emphasize surface-level and lexical features, while higher layers encode increasingly abstract semantic information (Tenney et al., 2019; Rogers et al., 2020; Clark et al., 2019). Recent studies on modern LLMs have refined this distinction, characterizing lower layers as memory units for lexical and factual retrieval (Wang et al., 2025; Yang et al., 2025), while higher layers perform abstract reasoning and manage complex subtasks (Yang et al., 2025). As a consequence, while focusing on token-level alignment may allow the student to reproduce teacher

*Equal contribution
†Corresponding author: linhnv@soict.hust.edu.vn

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32669–32684
July 2-7, 2026

activations at isolated layers, it fails to capture the progressive process through which representations are composed from lexical units into more abstract semantic structures, limiting generalization in downstream tasks.

To bridge this gap, we propose **Multi-Granular Trajectory Alignment (MTA)**, an extensible framework designed to align the evolutionary trajectory of representations between teacher and student models. Our approach is motivated by linguistic principles and supported by established findings in Transformer interpretability. These observations suggest that representations in LLMs evolve in a hierarchical, bottom-up manner: lower layers process fine-grained lexical dependencies (analogous to leaf nodes), whereas higher layers synthesize these inputs into abstract semantic structures (analogous to internal phrase nodes). Guided by this perspective, MTA adopts a layer-adaptive alignment strategy. Specifically, we introduce a Dynamic Structural Alignment ($\mathcal{L}_{DSA}$) objective that aligns fine-grained word-level spans at lower layers to ground lexical foundations, and coarser phrase-level spans (e.g., Noun Phrases, Verb Phrases) at higher layers to capture compositional semantics. By explicitly modeling these multi-granular relationships, MTA compels the student to mimic not just the teacher's final state, but the geometric dynamics of its information processing. Furthermore, we incorporate a Hidden Representation Alignment ($\mathcal{L}_{Hid}$) loss to ensure precise transfer via a weighted projection mechanism.

Our main contributions are summarized as follows:

- We identify the limitations of uniform intermediate alignment and propose **MTA**, a novel framework that aligns the representational trajectory of LLMs by leveraging their inherent hierarchical structure.
- We introduce a layer-adaptive distillation objective that transitions from word-level alignment in lower layers to phrase-level structural alignment in higher layers, reflecting the bottom-up compositional properties of language.
- We demonstrate that MTA serves as a generalized module that can be integrated into state-of-the-art distillation methods that share tokenization. Extensive experiments on standard instruction-following benchmarks show that MTA consistently boosts performance across diverse model architectures .

2 Background

This section introduces the background concepts underlying our approach. We first review the fundamentals of knowledge distillation for autoregressive language models, then discuss parse trees and feature dynamics distillation as perspectives that motivate our depth-wise alignment. An extended discussion of related work is deferred to Appendix B.

2.1 Knowledge Distillation Fundamentals

Knowledge Distillation (KD) (Hinton et al., 2015) transfers knowledge from a teacher to a student by minimizing the divergence between their conditional distributions. For autoregressive LMs, the sequence-distribution KL can be written as a sum over token-level conditionals; in practice it can be optimized under teacher forcing on a dataset (or sampled sequences), yielding a tractable token-wise KL objective. Specifically, for a dataset $\mathcal{D}$, the loss is aggregated at each time step t:

$$D_{\mathrm{KL}}(p, q_\theta) = \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{y}\sim p(\cdot|\mathbf{x})} \left[\log \frac{p(\mathbf{y}|\mathbf{x})}{q_\theta(\mathbf{y}|\mathbf{x})} \right] \quad (1)$$

$$\approx \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{t=1}^{|\mathbf{y}|} \sum_{y_t \in V} p(y_t|\mathbf{y}_{<t}, \mathbf{x}) \log \frac{p(y_t|\mathbf{y}_{<t}, \mathbf{x})}{q_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x})} \quad (2)$$

where V is the vocabulary and $\mathbf{y}_{<t}$ denotes the token history. This formulation enables the student to approximate the teacher's distribution and capture the inherent "dark knowledge" in inter-class correlations (Kim and Rush, 2016; Agarwal et al., 2024).

2.2 Parse Tree

A parse tree structurally represents the hierarchical nature of natural language, organizing flat text into meaningful syntactic units to capture semantic compositionality (Socher et al., 2011, 2013). In recursive deep learning, this topology guides the bottom-up merging of representations from leaf nodes to the root (Tai et al., 2015). Motivated by the hierarchical structure of language captured by parse trees, we treat the depth-wise evolution of Transformer representations as loosely analogous to bottom-up compositional construction, and design a layer-adaptive alignment strategy accordingly as shown in Figure 1.

2.3 Feature Dynamics Distillation

Feature Dynamics Distillation (FDD) (Gong et al., 2025) extends KD by viewing Transformer depth as discrete time steps of a continuous-depth dynamical system, following ODE-inspired interpretations of deep networks (Chen et al., 2018; Lu et al., 2019). Under this view, distillation should match not only intermediate states but also their layer-wise evolution. To handle dimension mismatch, FDD maps intermediate hidden states h_l through each model's LM head into the vocabulary space, denoted as $y_l = \log f_{\text{head}}(h_l)$. It then optimizes (i) a **Trajectory Loss** that aligns intermediate predictive distributions at selected layers, and (ii) a **Derivative Loss** that aligns finite-difference updates $\Delta y_l = y_l - y_{l-1}$ using cosine distance:

$$\mathcal{L}_{\text{Traj}} = \sum_j \mathcal{D}_{KL}\Big(P(y_{l_j}^{\mathcal{T}}) \,\|\, P(y_{l_j}^{\mathcal{S}})\Big), \quad (3)$$

$$\mathcal{L}_{\text{Der}} = \sum_j \Big(1 - \text{Cos}(\Delta y_{l_j}^{\mathcal{T}}, \Delta y_{l_j}^{\mathcal{S}})\Big). \quad (4)$$

While effective for capturing continuity in prediction dynamics, FDD aligns prediction dynamics using the same LM-head-based objective at the selected layers, without explicitly encoding depth-dependent linguistic structure or salience-aware token weighting (Gong et al., 2025). Inspired by the trajectory-based distillation framework of Gong et al. (2025), our approach departs from it by explicitly aligning the *evolution of relational geometry* across network depth. Rather than matching prediction trajectories alone, we leverage hierarchical structure to guide depth-aware alignment of token relations, enabling more semantically faithful knowledge transfer.

3 Methodology

In this section, we introduce **Multi-Granular Trajectory Alignment (MTA)**, a modular framework designed to enhance the distillation of Large Language Model (LLM). Unlike conventional methods that rely on uniform layer-to-layer mapping, MTA adopts a depth-aware alignment strategy that operates at different semantic granularities across network depth. MTA proposes two complementary objectives: Dynamic Structural Alignment ($\mathcal{L}_{\text{DSA}}$), which preserves the relational geometry among semantic units (words and phrases) within each layer, and Hidden Representation Alignment ($\mathcal{L}_{\text{Hid}}$), which enforces feature-level consistency between teacher and student representations. Together, these objectives constrain both the internal structure of representations within each layer and their transformation across depth.

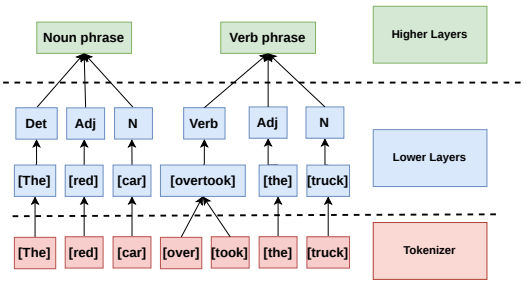

Figure 1: The correspondence between linguistic compositionality and the layer-wise evolution of representations in large language models.

3.1 Motivation: The Hierarchical Representational Trajectory

Most existing Knowledge Distillation (KD) methods match representations at fixed, static points, such as final logits or intermediate hidden states and attentions (Hinton et al., 2015; Sun et al., 2019; Jiao et al., 2019; Truong et al., 2025; Vuong et al., 2026; Xu et al., 2024). While effective for transferring local signals, these approaches largely ignore how representations evolve across network depth.

Interpretability studies consistently show that Transformer layers exhibit a hierarchical organization. Lower layers emphasize surface-level and lexical information, whereas higher layers progressively encode abstract semantic and compositional properties (Tenney et al., 2019; Rogers et al., 2020; Clark et al., 2019). Recent analyses of LLMs further characterize lower layers as repositories for factual and lexical memory (Wang et al., 2025), and higher layers as supporting abstract reasoning and compositional inference (Yang et al., 2025).

We therefore view a model's internal representations as forming a **hierarchical representational trajectory**: an ordered sequence of representation spaces whose semantic granularity systematically changes with depth. Under this perspective, applying a single, uniform alignment rule across all layers is suboptimal.

To respect this hierarchical trajectory during distillation, MTA uses *word-level spans* for alignment in lower layers to preserve lexical grounding, and *phrase-level spans* (e.g., noun and verb phrases) in higher layers to capture compositional structure.

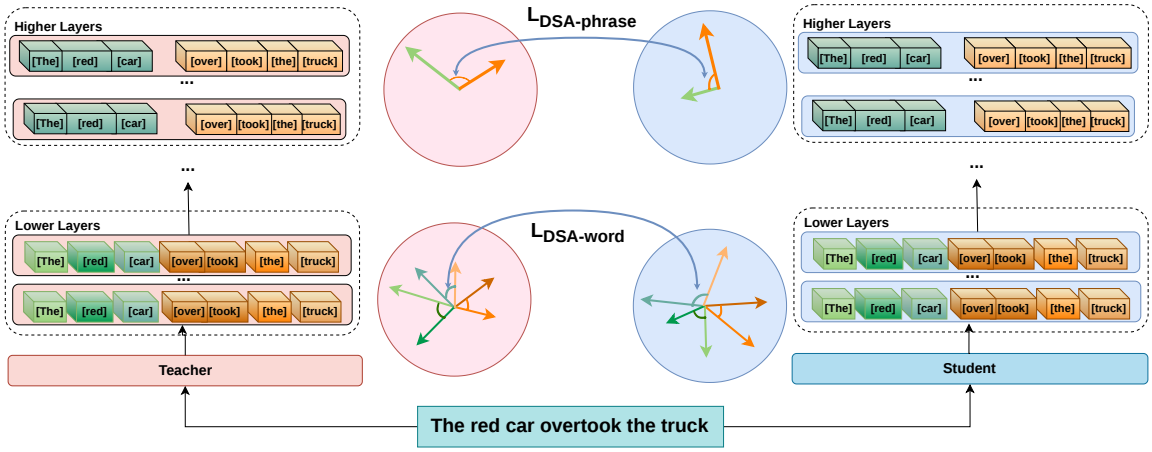

Figure 2: **Dynamic Structural Alignment ($\mathcal{L}_{\text{DSA}}$).** This objective enforces geometric consistency. It calculates the pairwise relational distances between semantic spans (words or phrases) within a layer for both Teacher and Student. By minimizing the discrepancy between these two topological structures across network depths, the Student learns to replicate the Teacher's representational trajectory.

This layer-adaptive design aims to better reflect the depth-wise transformation of representations than a single, uniform matching rule. Figure 1 illustrates the connection between linguistic compositionality and our depth-adaptive alignment strategy.

3.2 Layer-Adaptive Multi-Granular Spans

To operationalize this hierarchical alignment, we define semantic units tailored to different layers. We describe how we divide the selected key layers into *Lower* and *Higher* functional groups, detailed in Appendix D. To implement this, we utilize a syntactic parser (e.g., `spaCy`) to extract the corresponding spans from the input text.

Token Weight. Due to the unidirectional nature of causal attention in autoregressive LLMs, native attention weights are inherently biased toward earlier tokens, which tend to accumulate higher attention scores simply because they are visible to more subsequent positions when calculated based on attention centrality (or aggregated received attention). To bypass the unidirectional constraints and capture bidirectional dependencies, we compute token weights using a self-attention mechanism without self-loops. Let $\mathbf{H}_l \in \mathbb{R}^{N\times d}$ denote the hidden states at layer l. We first standardize token representations:

$$\hat{H}_{t,l} = \frac{H_{t,l}}{\sigma(H_{t,l})}. \tag{5}$$

Pairwise attention scores with diagonal and padding masking are computed as:

$$S_{s\to t,l} = \frac{\hat{H}_{s,l}\hat{H}_{t,l}^{\top}}{\sqrt{d}} + M_{s,t}. \tag{6}$$

The attention weights $\alpha_{s\to t,l}$ are then obtained by applying a softmax over the destination token dimension of the attention score matrix S_l. After that we compute the importance score of token t:

$$w_{t,l} = \frac{1}{N}\sum_{s=1}^{N}\alpha_{s\to t,l}. \tag{7}$$

These weights are used for span-level aggregation in subsequent objectives. The more detailed explanation is in Appendix C.

Span Definitions. At **lower layers**, we group tokens into complete *Word Spans* to capture fine-grained lexical processing. At **higher layers**, we extract *Noun Phrases (NPs)* and *Verb Phrases (VPs)* to serve as *Phrase Spans*, which are widely used as meaningful syntactic constituents in chunking and predicate-argument semantics (Ramshaw and Marcus, 1995; Gildea and Jurafsky, 2002). Spans are extracted over the entire input-output sequence, including both the prompt and the generated response. For each span k at layer l, we compute its representation $U_{k,l}$ as a weighted average of constituent token hidden states $H_{t,l}$:

$$U_{k,l} = \frac{\sum_{t\in S_k} w_{t,l}H_{t,l}}{\sum_{t\in S_k} w_{t,l}} \tag{8}$$

where S_k is the set of tokens in span k, and $w_{t,l}$ is the importance weight of token.

Span Weight. The weight of a span is computed by aggregating the weight of its constituent tokens and normalizing across spans. For span i at layer l, we define:

$$w_{i,l}^{\text{sp}} = \frac{\tilde{w}_{i,l}^{\text{sp}}}{\sum_{j=1}^{N_l^{\text{sp}}} \tilde{w}_{j,l}^{\text{sp}}}, \qquad \tilde{w}_{i,l}^{\text{sp}} = \sum_{t=s_i^l}^{e_i^l} w_{t,l}^{\mathcal{T}}. \tag{9}$$

where $[s_i^l, e_i^l]$ denotes the token indices covered by span i, $w_{t,l}^{T}$ is the Teacher-derived token importance weight, and N_l^{sp} is the number of spans at layer l.

3.3 Dynamic Structural Alignment Loss ($\mathcal{L}_{\text{DSA}}$)

The central objective of MTA is the **Dynamic Structural Alignment Loss**. This loss enforces geometric consistency between the Teacher and Student span representations, as shown in Figure 2. It operates on a set of selected key layers L_{key} (spanning both lower/word-level and higher/phrase-level layers). We formulate $\mathcal{L}_{\text{DSA}}$ to minimize the discrepancy in pairwise distances between spans within a layer:

$$\mathcal{L}_{\text{DSA}} = \frac{1}{\|L_{\text{key}}\|} \sum_{l \in L_{\text{key}}} \mathcal{L}_{\text{DSA}}^{(l)} \tag{10}$$

$$\mathcal{L}_{\text{DSA}}^{(l)} = \sum_{i=1}^{N_l^{\text{sp}}} \sum_{j=i+1}^{N_l^{\text{sp}}} w_{ij,\phi(l)}^{\text{sp}} \left(d(U_{i,l}^{\mathcal{S}}, U_{j,l}^{\mathcal{S}}) - d(U_{i,\phi(l)}^{\mathcal{T}}, U_{j,\phi(l)}^{\mathcal{T}}) \right)^2 \tag{11}$$

where $d(\cdot,\cdot)$ is cosine distance, N_l^{sp} is the number of spans at layer l and $\phi(\cdot)$ denote layer mapping function. We weight span pairs by salience, $w_{ij,\phi(l)}^{\text{sp}} = w_{i,\phi(l)}^{\text{sp}} w_{j,\phi(l)}^{\text{sp}}$, where $w_{k,\phi(l)}^{\text{sp}}$ aggregates token weights within span k. This prioritizes alignment between semantically salient spans. Minimizing $\mathcal{L}_{\text{DSA}}$ encourages the student to match the teacher's within-layer relational structure and its evolution across depth.

3.4 Hidden Representation Alignment ($\mathcal{L}_{\text{Hid}}$)

To complement the structural constraints, we incorporate a feature-level loss to align specific hidden states. Since the Student dimension ($d_{\mathcal{S}}$) is typically smaller than the Teacher dimension ($d_{\mathcal{T}}$), we employ a learnable linear projector $W_l \in \mathbb{R}^{d_{\mathcal{S}} \times d_{\mathcal{T}}}$ to map the Student's representations into the Teacher's space:

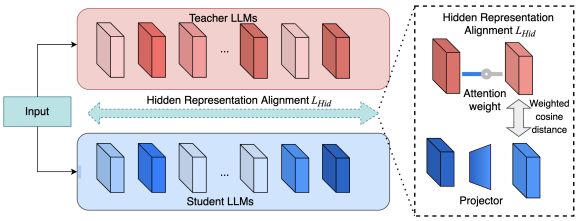

Figure 3: **Hidden Representation Alignment Strategy.** The student learns to match teacher representations using a weighted cosine distance objective, ensuring accurate feature reconstruction at key layers.

$$\tilde{H}_{t,l}^{\mathcal{S}} = H_{t,l}^{\mathcal{S}} W_l \tag{12}$$

We then minimize the weighted cosine distance between the projected Student states and the Teacher states:

$$\mathcal{L}_{\text{Hid}} = \sum_{l \in \mathcal{L}_{\text{key}}} \sum_{t \in \mathcal{M}_l} w_{t,l}^{\mathcal{T}} \left(1 - \frac{\langle \tilde{H}_{t,l}^{\mathcal{S}}, H_{t,l}^{\mathcal{T}} \rangle}{\left\| \tilde{H}_{t,l}^{\mathcal{S}} \right\|_2 \left\| H_{t,l}^{\mathcal{T}} \right\|_2} \right) \tag{13}$$

where L_{key} denotes the set of selected intermediate layers used for distillation, $\mathcal{M}_l$ contains tokens covered by the extracted spans at layer l, and $w_{t,l}^{\mathcal{T}}$ is the Teacher-derived token importance weight defined previously. This ensures that the Student focuses on replicating the features of the most informative tokens. Figure 3 provides an overview of how the proposed loss works.

3.5 Optimization Objective

MTA is designed as a modular plug-in for existing distillation frameworks (e.g., DistiLLM (Ko et al., 2024), DistiLLM-2 (Ko et al., 2025), FDD (Gong et al., 2025)). The total training objective combines the base distillation loss $\mathcal{L}_{\text{Base}}$ of the original methods with our proposed structural and hidden losses:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Base}} + \lambda_{\text{DSA}} \mathcal{L}_{\text{DSA}} + \lambda_{\text{Hid}} \mathcal{L}_{\text{Hid}} \tag{14}$$

where λ_{DSA} and λ_{Hid} are hyperparameters balancing structural preservation and feature alignment.

4 Experiments

4.1 Experimental Setup

Datasets. Our evaluation spans multiple instruction-following datasets. We adopt the

preprocessing procedure of (Zhang et al., 2024). Distillation is trained on DATABRICKS-DOLLY-15K. For evaluation, we report ROUGE-L on the Dolly test split and on four out-of-distribution benchmarks - S-NI (Wang et al., 2022), VICUNAEVAL (Chiang et al., 2023), and SELFINST (Wang et al., 2023b).

Training and Evaluation Settings. We evaluate our method on three student models: GPT-2-120M (Radford et al., 2019), Qwen1.5-0.5B (Bai et al., 2023), and OPT-1.3B (Zhang et al., 2022). For each student model, we use a correspondingly larger teacher model of the same family: GPT-2-1.5B, Qwen1.5-1.8B, and OPT-6.7B. To assess the quality of the generated outputs, we employ the ROUGE-L score (Lin, 2004) as our primary evaluation metric. Additional training and evaluation setup details are provided in Appendix D.

Baselines. We benchmark our method against several prominent frameworks that employ the same tokenizer, including FDD (Gong et al., 2025), DistiLLM (Ko et al., 2024), and DistiLLM-2 (Ko et al., 2025). A detailed description of FDD is provided in Section 2.3, while overviews of DistiLLM and DistiLLM-2 are presented in Appendix A. We integrate our proposed method into these baselines to evaluate its effectiveness as a complementary optimization strategy.

4.2 Main Results

Table 1 provides an overview of the ROUGE-L evaluation results for all teacher-student configurations examined in this study that range from lightweight to large-scale architectures. As observed, the integration of our MTA module consistently improves the performance of all baseline frameworks, effectively narrowing the performance gap between student and teacher models. This improvement is robust across varying scales and architectures, including GPT-2, Qwen1.5, and OPT. Such scalability suggests that our approach functions as a generalized plug-and-play optimization layer.

Beyond ROUGE-L–based metrics, we employ GPT-4o-mini as an evaluator (LLM-as-a-judge) using the prompts detailed in Appendix F to assess the semantic quality of the generated outputs. The model is prompted to score responses based on factual accuracy, coherence, and helpfulness. As illustrated in Figure 4 and Figure 5,

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg.
GPT-2 1.5B → GPT-2 120M					
Teacher	28.71	15.68	17.00	28.80	22.55
SFT	23.33	10.56	15.12	17.08	16.52
FDD	25.47	12.45	16.44	23.54	19.48
w/ MTA	25.64	13.6	17.00	25.75	20.50
DistiLLM	25.65	13.39	16.50	25.28	20.21
w/ MTA	25.77	14.19	16.67	29.18	21.45
DistiLLM-2	22.44	12.52	12.30	27.10	18.59
w/ MTA	24.76	14.16	14.28	26.54	19.94
Qwen1.5 1.8B → Qwen1.5 0.5B					
Teacher	28.23	19.58	19.59	34.36	25.44
SFT	24.83	13.31	16.97	22.07	19.30
FDD	25.08	12.24	16.08	23.69	19.27
w/ MTA	25.35	13.85	17.21	27.25	20.92
DistiLLM	25.16	12.90	15.86	25.28	19.80
w/ MTA	25.61	13.08	16.04	29.32	21.01
DistiLLM-2	27.48	17.95	17.14	30.99	23.39
w/ MTA	27.93	18.87	18.63	33.49	24.73
OPT 6.7B → OPT 1.3B					
Teacher	27.60	16.40	17.80	30.30	23.03
SFT	26.00	11.40	15.60	23.10	19.03
FDD	26.07	14.82	17.09	28.98	21.74
w/ MTA	26.49	16.47	17.51	31.12	22.90
DistiLLM	27.87	15.99	18.02	30.02	22.98
w/ MTA	27.43	17.07	18.40	32.97	23.97
DistiLLM-2	26.31	17.62	17.60	30.85	22.96
w/ MTA	26.79	18.16	16.55	31.39	23.22

Table 1: Performance of different distillation methods across datasets. "w/ MTA" denotes incorporating our proposed Multi-Granular Trajectory Alignment (MTA) strategy on top of each baseline.

our method yields consistent improvements in evaluation scores across both architectures. Notably, in the FDD framework, the integration of MTA leads to significant gains. Even for stronger baselines like DistiLLM and DistiLLM-2, our approach maintains a positive trajectory. These results confirm that the performance benefits of MTA are architecture-agnostic, effectively enhancing the semantic correctness and helpfulness of diverse student models.

5 Analysis

To assess the contribution of individual components in MTA, we conduct ablation studies using the *GPT-2 1.5B → GPT-2 120M* distillation setting. We analyze two key aspects: (1) the effect

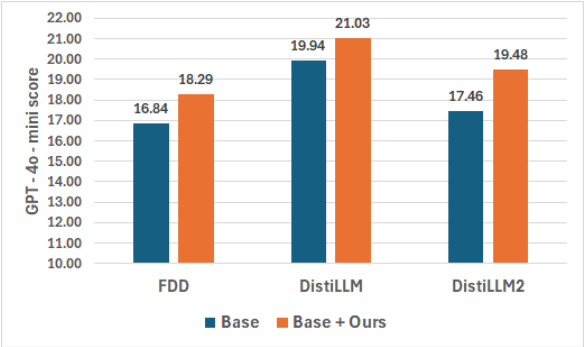

Figure 4: GPT-4o-mini evaluation scores (1-100) for distilling GPT-2 1.5B into GPT-2 120M.

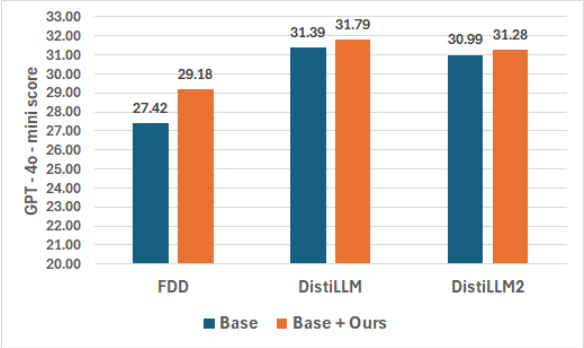

Figure 5: GPT-4o-mini evaluation scores (1-100) for distilling OPT 6.7B into OPT 1.3B.

of each loss component, and (2) the impact of our layer-adaptive multi-granular alignment strategy. Additional results are reported in Appendix E.

5.1 Impact of Loss Components

Table 2 evaluates the Dynamic Structural Alignment loss ($\mathcal{L}_{\text{DSA}}$) and the Hidden Representation Alignment loss ($\mathcal{L}_{\text{Hid}}$) when added to FDD, DistiLLM, and DistiLLM-2.

Effect of Individual Losses. We observe that adding either $\mathcal{L}_{\text{Hid}}$ or $\mathcal{L}_{\text{DSA}}$ individually improves the average performance across all baselines compared to the corresponding baseline. Adding $\mathcal{L}_{\text{Hid}}$ yields moderate gains by directly aligning intermediate features, improving the average score across all frameworks (e.g., +0.34 for DistiLLM). Introducing $\mathcal{L}_{\text{DSA}}$ often provides larger gains, particularly on benchmarks such as S-NI, indicating that preserving relational structure among semantic units offers complementary supervision beyond point-wise feature matching.

Synergy of the Full Method. Using both losses together achieves the strongest results across all baselines. For example, under DistiLLM, the full MTA configuration improves the average score from 20.21 to 21.45. These results suggest that structural alignment and feature-level alignment capture complementary aspects of the distillation process and are most effective when applied jointly.

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg.
GPT-2 1.5B → GPT-2 120M					
FDD	25.47	12.45	16.44	23.54	19.48
w/ $\mathcal{L}_{\text{Hid}}$	25.39	12.42	16.9	24.18	19.72
w/ $\mathcal{L}_{\text{DSA}}$	25.44	12.47	17.08	24.49	19.87
w/ Full	25.64	13.6	17.00	25.75	20.50
DistiLLM	25.65	13.39	16.50	25.28	20.21
w/ $\mathcal{L}_{\text{Hid}}$	25.89	13.68	16.86	25.77	20.55
w/ $\mathcal{L}_{\text{DSA}}$	25.77	14.24	16.27	27.40	20.92
w/ Full	25.77	14.19	16.67	29.18	21.45
DistiLLM-2	22.44	12.52	12.30	27.10	18.59
w/ $\mathcal{L}_{\text{Hid}}$	25.26	13.13	13.53	25.79	19.43
w/ $\mathcal{L}_{\text{DSA}}$	25.08	13.28	14.09	27.06	19.88
w/ Full	24.76	14.16	14.28	26.54	19.94

Table 2: Performance of loss combinations on instruction-following benchmarks.

5.2 Impact of Hierarchical Granularity

Table 3 investigates the validity of our core hypothesis: that distillation should be layer-adaptive, aligning *Word spans* at lower layers and *Phrase spans* at higher layers. We compare our adaptive approach (*Full-level*) against static strategies that enforce uniform granularity (Word-only or Phrase-only) across all layers.

Using a single granularity throughout the network leads to suboptimal performance. A **word-level-only strategy** is effective for tasks dominated by surface patterns, but it consistently underperforms on reasoning-intensive benchmarks (e.g., Vicuna). For instance, the word-level variant of DistiLLM-2 achieves an average score of 19.10, substantially lower than the adaptive approach. This suggests that enforcing fine-grained alignment at higher layers can restrict the student's ability to form abstract representations. In contrast, a **phrase-level-only** strategy generally yields stronger results than word-level alignment (e.g., 21.17 vs. 20.80 for DistiLLM), supporting the view that semantic constituents carry richer information. Nevertheless, it still underperforms relative to the full adaptive method, likely because ignoring word-level grounding at lower layers weakens the model's lexical foundation.

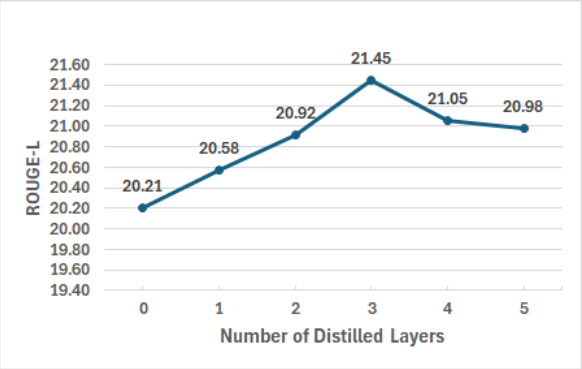

Figure 6: Effect of the number of distilled intermediate layers. Averaged ROUGE-L scores using DistiLLM + our approach.

Superiority of Trajectory Alignment. The *Full-level* (Layer-Adaptive) strategy consistently achieves the highest scores across all baselines (e.g., 20.50 for FDD and 21.45 for DistiLLM). Notably, on the S-NI benchmark, the adaptive approach in DistiLLM reaches 29.18, outperforming the Phrase-only variant by nearly 2 points. This empirically validates our motivation: aligning the student's representational trajectory to match the teacher's functional hierarchy - moving from lexical grounding to semantic composition - is crucial for distilling capable and robust language models. Furthermore, to assess the contribution of span weights, we conduct additional ablation experiments, the results of which are reported in Appendix E.

Methods	Dolly	SelfInst	Vicuna	S-NI	Avg.
GPT-2 1.5B → GPT-2 120M					
FDD	25.47	12.45	16.44	23.54	19.48
w/ Word-level	25.01	12.82	17.53	24.99	20.09
w/ Phrase-level	25.29	12.87	17.36	24.55	20.02
w/ Full-level	25.64	13.60	17.00	25.75	20.50
DistiLLM	25.65	13.39	16.50	25.28	20.21
w/ Word-level	25.82	13.54	16.67	27.16	20.80
w/ Phrase-level	25.96	14.25	17.03	27.42	21.17
w/ Full-level	25.77	14.19	16.67	29.18	21.45
DistiLLM-2	22.44	12.52	12.30	27.10	18.59
w/ Word-level	25.05	12.87	14.06	24.41	19.10
w/ Phrase-level	24.70	13.40	14.14	25.35	19.40
w/ Full-level	24.76	14.16	14.28	26.54	19.94

Table 3: Ablations on hierarchical level mechanisms.

5.3 Sensitivity Analysis on Layer Selection

This section analyzes the sensitivity of the distillation process to the number of aligned layers (M) and their specific granularity allocation.

Config	W:P	Dolly	SelfInst	Vicuna	S-NI	Avg.
All Phrase	0 : 3	25.96	14.25	17.03	27.42	21.17
All Word	3 : 0	25.82	13.54	16.67	27.16	20.80
Hybrid A	2 : 1	25.89	14.02	16.38	27.30	20.90
MTA (Ours)	**1 : 2**	25.77	14.19	16.67	29.18	**21.45**

Table 4: Impact of granularity allocation under the DistiLLM framework with our proposed method. We vary the ratio of Word-level (W) to Phrase-level (P) spans and report performance across benchmarks.

Effect of the number of distilled intermediate layers. Figure 6 illustrates the impact of increasing the layer budget M on the student's performance (Avg. ROUGE-L). We observe an inverted U-shaped trend. Performance improves significantly as we increase from $M = 0$ (baseline) to $M = 3$, reaching a peak score of 21.45. This confirms that sparse intermediate supervision provides necessary guidance for the student's trajectory. However, further increasing M to 4 or 5 leads to diminishing returns and a slight performance drop. This decline supports the hypothesis regarding *inter-layer redundancy*. Thus, we select $M = 3$ as the optimal budget for the GPT-2 pair; deeper models use proportionally larger M following the same strided rule (see Table 11).

Word vs. Phrase Allocation. Fixing the budget at $M = 3$, we further analyze how to allocate these layers between Word-level and Phrase-level alignment (Table 4). Purely structural alignment (*All Phrase*) outperforms purely lexical alignment (*All Word*), highlighting the importance of semantic compositionality. However, the best performance is achieved with a **Hybrid** configuration (1 Word : 2 Phrase). This empirically validates our hypothesis that the lowest selected layer should be dedicated to lexical grounding, while the upper layers specialize in semantic abstraction.

Robustness of Layer Schedule. To further validate that MTA's improvements are not sensitive to a specific hand-picked layer configuration, we conduct a robustness study on the Qwen1.5 pair (Qwen1.5-1.8B → Qwen1.5-0.5B) under multiple word/phrase layer schedules. As described in Section D, supervision points are selected via a deterministic strided top-down rule and mapped to teacher layers by proportional scaling, without any per-run search. Table 5 reports results across five

alternative schedules, all derived from the same deterministic rule with different stride or budget choices.

Across all configurations, MTA consistently improves over the DistiLLM baseline (Avg. ROUGE-L: 19.80), with gains ranging from +0.75 to +1.21 points. Performance variation across schedules is small (typically $<$0.5 points), confirming that MTA does not rely on a single carefully tuned configuration. In practice, the default strided top-down schedule (Section D) is sufficient to reliably improve distillation quality without extensive hyperparameter search. Crucially, no per-run search over word/phrase assignments is needed: our empirical finding that assigning the *lowest* selected layer to word-level and all remaining layers to phrase-level is a robust default that performs competitively across all tested configurations.

Word-level	Phrase-level	Dolly	SelfInst	Vicuna	S-NI	Avg.
DistiLLM (baseline)		25.16	12.90	15.86	25.28	19.80
14	16–24	25.61	13.08	16.04	**29.32**	**21.01**
14, 16, 18	20, 22, 24	25.30	13.02	16.05	29.10	20.87
16, 18, 20	22, 24	25.51	**13.41**	**16.27**	27.02	20.55
19	20–24	**26.40**	14.63	16.08	25.54	20.66
12, 14, 16	18–24	25.11	13.11	16.50	29.17	20.97

Table 5: Sensitivity of MTA to different layer schedule configurations on Qwen1.5-1.8B $\rightarrow$ Qwen1.5-0.5B. Here we only use even layers.

5.4 Computational Efficiency Analysis

Table 6 reports the per-step wall-clock time and GPU memory consumption of MTA compared to the baselines. All experiments are conducted on a single NVIDIA A100 (40 GB) GPU with a batch size of 16. The additional training overhead in MTA stems from the syntactic span extraction stage (via spaCy) and the span-level geometry matching in $\mathcal{L}_{\text{DSA}}$.

In our initial implementation, span extraction contributed 0.42 s/step, resulting in a 2.54$\times$ slowdown over DistiLLM. After optimizing the extraction pipeline via larger spaCy batch sizes, this overhead is reduced to 0.27 s/step, bringing the overall slowdown down to 1.85$\times$ with no increase in GPU memory usage. Importantly, **MTA adds overhead only during training**; inference cost remains completely unchanged since span extraction is not used at test time.

To verify that the performance gains are not merely a consequence of increased training budget, we conduct a time-matched comparison where the baselines are trained for 3$\times$ more epochs, granting them a wall-clock budget equal to or exceeding that of MTA. As shown in Table 7, even with substantially more training time, the baselines do not close the performance gap, confirming that MTA's improvements stem from the proposed trajectory-alignment objective rather than extra compute.

Method	Span Extr.	Time	avg_alloc	peak_alloc
DistiLLM	0.00	0.26	6.53	16.91
DistiLLM + MTA	0.27	0.48	6.54	17.94
FDD	0.00	0.49	6.67	23.04
FDD + MTA	0.36	0.69	6.70	24.05

Table 6: Computation time and GPU memory consumption. MTA introduces overhead only during training; inference cost is unchanged.

Training Setup	Dolly	SelfInst	Vicuna	S-NI	Avg.
DistiLLM variants					
Train w/ 5 ep	25.65	13.39	16.50	25.28	20.21
Train w/ 10 ep	26.64	13.11	17.49	23.83	20.27
Train w/ 15 ep	26.61	13.35	16.96	24.13	20.26
w/ MTA (5 ep)	**25.77**	**14.19**	**16.67**	**29.18**	**21.45**
DistiLLM-2 variants					
Train w/ 5 ep	22.44	12.52	12.30	27.10	18.59
Train w/ 10 ep	22.96	12.51	12.88	27.91	19.07
Train w/ 15 ep	22.97	12.46	13.09	27.68	19.05
w/ MTA (5 ep)	**24.76**	**14.16**	**14.28**	**26.54**	**19.94**

Table 7: Time-matched comparison on GPT-2 1.5B $\rightarrow$ GPT-2 120M. Baselines trained for up to 15 epochs (3$\times$ the wall-clock budget of MTA at 5 epochs) do not close the performance gap.

6 Conclusion

We introduce **Multi-Granular Trajectory Alignment (MTA)**, a layer-adaptive distillation framework that models the hierarchical representational trajectory of large language models. MTA aligns word-level representations in lower layers and phrase-level representations in higher layers, matching the depth-dependent semantic roles of Transformer layers. By combining structural and hidden-state alignment objectives, MTA enables students to preserve both local features and global relational geometry, yielding consistent gains across strong baselines.

7 Limitations

While MTA improves distillation quality, it introduces additional computational cost due to the use of external sparse structures for layer-wise alignment. Although this overhead remains manageable in our current experiments, an important direction for future work is to design more lightweight yet still layer-adaptive alignment mechanisms that reduce computational cost while preserving performance. In addition, our experiments are conducted under fixed computational budgets and benchmark settings. We view these constraints as opportunities for further development, particularly in exploring more efficient layer-wise alignment designs and adaptive mapping strategies that can enhance the scalability and robustness of MTA.

Acknowledgments

This project was supported by the Air Force Office of Scientific Research under award number FA9550-23-S-0001.

References

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

Anshumann Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. 2025. Sparse Logit Sampling: Accelerating Knowledge Distillation in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18085–18108, Vienna, Austria. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. 2018. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

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. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, MA.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286. Association for Computational Linguistics.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8592–8600.

Stephen Crain and Mineharu Nakayama. 1987. Structure dependence in grammar formation. *Language*, pages 522–543.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Guoqiang Gong, Jiaxing Wang, Jin Xu, Deping Xiang, Zicheng Zhang, Leqi Shen, Yifeng Zhang, JunhuaShu JunhuaShu, ZhaolongXing ZhaolongXing, Zhen Chen, et al. 2025. Beyond logits: Aligning feature dynamics for effective knowledge distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23067–23077.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. 2025. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint arXiv:2503.07067*. ICML 2025 Spotlight.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*. ICML 2024.

Tue Le, Hoang Tran Vuong, Quyen Tran, Linh Ngo Van, Mehrtash Harandi, and Trung Le. Token-level self-play with importance-aware guidance for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*.

Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026. CTPD: cross tokenizer preference distillation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI*, pages 37783–37790. AAAI Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 129–136.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Association for Computational Linguistics.

Xinyuan Song, Keyu Wang, PengXiang Li, Lu Yin, and Shiwei Liu. 2025. Demystifying the roles of llm layers in retrieval, knowledge, and reasoning.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1556–1566. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics.

Minh-Phuc Truong, Hai An Vu, Tu Vu, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2025. Emo: Embedding model distillation via intra-model relation and optimal transport alignments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7605–7617.

Duc Trung Vu, Pham Khanh Chi, Dat Phi Van, Linh Ngo Van, Dinh Viet Sang, and Trung Le. 2026a. Dwa-kd: Dual-space weighting and time-warped alignment for cross-tokenizer knowledge distillation. In *Findings of the Association for Computational Linguistics: EACL*, pages 3513–3527.

Hai An Vu, Minh-Phuc Truong, Tu Vu, and Linh Ngo. 2026b. Mol: Mixture of layers in cross-tokenizer embedding model distillation. *Knowledge-Based Systems*, 343:116001.

Hoang Tran Vuong, Tue Le, Quyen Tran, Linh Ngo Van, and Trung Le. 2026. MCW-KD: multi-cost wasserstein knowledge distillation for large language models. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI*, pages 33332–33340. AAAI Press.

Jinke Wang, Zenan Ying, Qi Liu, Wei Chen, Tong Xu, Huijun Hou, and Zhi Zheng. 2025. Think and recall: Layer-level prompting for lifelong model editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14498–14513, Suzhou, China. Association for Computational Linguistics.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023a. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-Instruct: Aligning language models with Self-Generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. BERT-of-theseus: Compressing BERT by progressive module replacing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7859–7869, Online. Association for Computational Linguistics.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models.

Zhipeng Yang, Junzhuo Li, Siyu Xia, and Xuming Hu. 2025. Internal chain-of-thought: Empirical evidence for layer-wise subtask scheduling in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22547–22575, Suzhou, China. Association for Computational Linguistics.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

A Baseline models

A.1 DistiLLM

To address the optimization instability and high computational costs associated with standard KD and on-policy data generation, Ko et al. (2024) proposed DistiLLM. This framework introduces two primary innovations:

Skew (Reverse) Kullback-Leibler Divergence: Standard KLD often leads to optimization instability due to gradient explosion when the student model assigns low probability to the teacher's support. To mitigate this, DistiLLM utilizes **Skew KLD**, which interpolates the target distribution with the student's distribution. Formally, the α-Skew KLD is defined as:

$$D_{SKL}^{(\alpha)}(p, q_\theta) = D_{KL}(p \parallel \alpha p + (1-\alpha) q_\theta) \quad (15)$$

Similarly, to leverage the mode-seeking behavior of Reverse KLD while maintaining gradient stability, they proposed **Skew Reverse KLD (SRKL)**:

$$D_{SRKL}^{(\alpha)}(p, q_\theta) = D_{KL}(q_\theta \parallel (1-\alpha) p + \alpha q_\theta) \quad (16)$$

where p is the teacher, q_θ is the student, and $\alpha \in [0, 1]$ controls the mixing ratio. These modifications prevent the denominator in the gradient term from vanishing, thereby bounding the gradient norm and ensuring smoother convergence.

Adaptive Off-Policy Generation: While utilizing Student-Generated Outputs (SGOs) addresses training-inference mismatch, generating sequences at every iteration (on-policy) is computationally expensive. DistiLLM proposes an efficient off-policy strategy using a replay buffer $\mathcal{D}_R$ to store and reuse SGOs. Furthermore, it employs an *adaptive SGO scheduler* that dynamically adjusts the probability ϕ of sampling SGOs based on the student's validation loss trends. This mechanism ensures SGOs are introduced only when necessary to correct distribution shifts, balancing training efficiency with the mitigation of noisy feedback.

A.2 DistiLLM-2

Building upon the success of Skew KLD and adaptive data strategies, Ko et al. (2025) introduced DistiLLM-2, a contrastive framework designed to align teacher and student models more effectively across varied data types. The method comprises three key technical advancements:

Contrastive Approach for LLM Distillation (CALD): Motivated by the observation that Forward KL and Reverse KL exhibit asymmetric behaviors—"pulling up" probabilities on teacher-generated outputs (TGOs) and "pushing down" on student-generated outputs (SGOs), respectively—DistiLLM-2 proposes a dual-loss objective. It applies Skew KL (SKL) to TGOs (y_t) to encourage matching high-probability regions, and Skew Reverse KL (SRKL) to SGOs (y_s) to suppress low-quality generations. The combined objective is defined as:

$$\mathcal{L}_{CALD} = \frac{1}{2|\mathcal{D}|} \sum_{(x, y_t, y_s) \sim \mathcal{D}} \Big[(1-\beta) D_{SKL}^{(\alpha_t)}(x, y_t) + \beta D_{SRKL}^{(\alpha_s)}(x, y_s) \Big] \quad (17)$$

where β balances the two terms. This formulation is mathematically connected to preference optimization methods like DPO but tailored for distillation to avoid reward hacking.

Curriculum-based Adaptive Learning: To further refine the objective, DistiLLM-2 introduces dynamic updates for the skew coefficient α. Recognizing that the optimal mixing ratio depends on the sample difficulty (the gap between teacher and student distributions), the method employs a curriculum-based update rule derived from a first-order Taylor approximation:

$$\alpha \approx 1 - (1 - \alpha_0) \frac{m}{p(y|x) - q_\theta(y|x)} \quad (18)$$

Additionally, the coefficient β for the SRKL term is gradually increased during training, shifting focus from imitating the teacher to correcting student errors as the model capability improves.

Optimal Data Curation: Instead of purely on-policy generation, DistiLLM-2 adopts a "batched" data curation strategy where SGOs are collected ahead of each training epoch. This approach maintains the benefits of on-policy feedback while significantly improving computational efficiency and compatibility with high-throughput inference engines like vLLM.

B Extended Related Work

Knowledge distillation for language models. Knowledge distillation (KD) trains a compact student to imitate a larger teacher, most commonly by matching the teacher's output distribution via KL-based objectives (Hinton et al., 2015; Kim and Rush, 2016). In the context of large language models (LLMs), logit-level KD remains a strong baseline, but can suffer from optimization instability

and train-test mismatch when trained purely under teacher forcing (Agarwal et al., 2024). Recent frameworks improve practicality and stability through better divergence formulations and data strategies (Ko et al., 2024, 2025; Agarwal et al., 2024; Anshumann et al., 2025). These methods primarily focus on aligning behavior at the output level, which is effective but may under-constrain the internal knowledge encoded in intermediate representations.

Intermediate representation alignment and trajectory-based distillation. A complementary line of work distills internal signals - hidden states, attention maps, and other intermediate features - to transfer information beyond output logits. Representative Transformer distillation methods align intermediate hidden states and/or attentions (Jiao et al., 2019; Sun et al., 2019, 2020), while others emphasize distilling attention relations to build smaller but competitive students (Wang et al., 2020). Recent work also revisits intermediate supervision for LLM distillation under efficiency constraints, often choosing a sparse subset of layers (Gong et al., 2025; Song et al., 2025; Vu et al., 2026b).

Beyond matching isolated layers, trajectory-style methods aim to match how representations change across depth. For example, Feature Dynamics Distillation (FDD) explicitly treats intermediate features as a discretized depth-wise path and aligns not only the feature trajectory but also its first-order dynamics (Gong et al., 2025). Trajectory ideas also appear at the training-process level: BERT-of-Theseus progressively replaces teacher modules with compact substitutes during training (Xu et al., 2020). Separately, some work studies "trajectory" over generated tokens by distilling step-by-step rationales (Wang et al., 2023a; Li et al., 2023).

Hierarchy in Transformer representations and linguistically motivated compression. A substantial body of work shows that Transformer layers form a functional hierarchy, where lower layers encode surface/lexical information and higher layers capture increasingly abstract and semantic features (Tenney et al., 2019; Rogers et al., 2020; Clark et al., 2019). Linguistics similarly characterizes language as hierarchical composition from lexical units into phrases and higher-level structures (Chomsky, 1965; Crain and Nakayama, 1987; Socher et al., 2011, 2013). Recent analyses further suggest that modern LLMs exhibit depth-specialized behaviors such as memory-like retrieval in lower layers and more abstract reasoning in higher layers (Wang et al., 2025; Yang et al., 2025). In addition, representational *geometry* has been shown to encode separable syntactic/semantic structure and fine-grained organization of linguistic features in BERT (Coenen et al., 2019), further motivating distillation objectives that preserve relational structure rather than only per-token values. These findings motivate distillation objectives that respect hierarchical structure. In our work, we instantiate this idea as *Multi-Granular Trajectory Alignment (MTA)*, which performs depth-adaptive alignment: word-level units for lower layers and phrase-level spans (e.g., NPs/VPs) for higher layers, thereby constraining not only feature values but also the evolving relational geometry across depth.

MTA complements both logit-based and intermediate-representation KD by adding (i) *Dynamic Structural Alignment* that matches within-layer relational structure among semantic units, and (ii) *Hidden Representation Alignment* for selected layers, with salience-aware weighting. Unlike depth-uniform feature matching, our layer-adaptive design explicitly tracks the hierarchical representational trajectory, aligning lexical grounding early and compositional semantics later, and can be plugged into recent LLM distillation pipelines (Ko et al., 2024, 2025; Gong et al., 2025).

C Token Importance Computation

To compute the token-level importance weights $w \in \mathbb{R}^N$, we employ a standardized pairwise self-attention mechanism that captures the global contextual relevance of each token while explicitly excluding self-loops. Let $\mathbf{H}_l = [\mathbf{h}_{1,l}, \ldots, \mathbf{h}_{N,l}] \in \mathbb{R}^{N \times d}$ denote the sequence of hidden states at layer l-th, where N is the sequence length and d is the feature dimension.

First, to stabilize the dot-product scores, we standardize the hidden states to obtain $\hat{\mathbf{H}}$, where each vector is scaled by its standard deviation:

$$\hat{H}_{t,l} = \frac{H_{t,l}}{\sigma(H_{t,l})}, \tag{19}$$

where $\sigma(\cdot)$ computes the standard deviation along the feature dimension.

Next, we calculate the pairwise attention scores $\mathbf{S}_l \in \mathbb{R}^{N \times N}$. We incorporate a masking term $\mathbf{M}$ to

enforce two constraints: (1) tokens cannot attend to padding elements, and (2) tokens cannot attend to themselves (diagonal masking), forcing the model to rely solely on context:

$$S_{s\to t,l} = \frac{\hat{H}_{s,l}\hat{H}_{t,l}^{\top}}{\sqrt{d}} + M_{s,t}, \tag{20}$$

where the mask $M_{s,t}$ enforces both padding and diagonal constraints:

$$M_{s,t} = \begin{cases} -\infty, & s = t \text{ or } t \in \mathcal{P}, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where $\mathcal{P}$ represents the set of padding indices.

The attention weights are obtained via the softmax function. Finally, the scalar importance weight w_t for the t-th token is computed as the mean attention it receives from all other tokens in the sequence (column-wise average):

$$\alpha_{s\to t,l} = \frac{\exp(S_{s\to t,l})}{\sum_{u=1}^{N}\exp(S_{s\to u,l})}. \tag{22}$$

$$w_{t,l} = \frac{1}{N}\sum_{s=1}^{N}\alpha_{s\to t,l}. \tag{23}$$

The resulting vector $\mathbf{w}_l = [w_{1,l}, \ldots, w_{N,l}]$ represents the computed token weights at layer l-th.

D Experimental Details

Training and Evaluation For GPT-2 and Qwen1.5 models, we perform full-parameter fine-tuning, while for OPT models we adopt LoRA-based parameter-efficient fine-tuning. Detailed training configurations for each model are summarized in Table 8 and Table 9. For evaluation, we sample model outputs using five different random seeds to account for stochasticity in generation. The final performance is reported as the average ROUGE-L score (Lin, 2004) between the generated responses and the human-annotated references.

Hyperparameter The hyperparameters, λ_{DSA} and λ_{Hid}, are reported in Table 10. All reported values are selected based on preliminary validation experiments.

Layer Mapping Configuration. Table 11 summarizes the selected intermediate layers used for distillation at the word and phrase levels. For models with different depths, we align student layers to teacher layers using a fixed integer mapping.

Settings	**GPT2-1.5B**	**Qwen1.5-1.8B**	**OPT-6.7B**
	GPT2-120M	Qwen1.5-0.5B	OPT-1.3B
Epoch	5	5	5
LR	1×10^{-4}	1×10^{-4}	5×10^{-4}
Projector LR	5×10^{-4}	5×10^{-4}	5×10^{-4}
Batch Size	16	16	16
LR Scheduler	Cosine	Cosine	Cosine
Fine-Tuning	Full	Full	LoRA
LoRA Rank	–	–	16
LoRA Alpha	–	–	64
LoRA Dropout	–	–	0.1

Table 8: Training configurations for FDD and DistiLLM.

Settings	**GPT2-1.5B**	**Qwen1.5-1.8B**	**OPT-6.7B**
	GPT2-120M	Qwen1.5-0.5B	OPT-1.3B
Epoch	5	5	3
LR	1×10^{-4}	5×10^{-5}	5×10^{-4}
Projector LR	5×10^{-4}	5×10^{-4}	5×10^{-4}
Batch Size	8	8	8
LR Scheduler	Cosine	Cosine	Cosine
Fine-Tuning	Full	Full	LoRA
LoRA Rank	–	–	16
LoRA Alpha	–	–	128
LoRA Dropout	–	–	0.1

Table 9: Training configurations for DistiLLM-2.

Motivated by the high inter-layer redundancy observed in Transformers (Song et al., 2025; Gong et al., 2025), where adjacent layers often encode similar features, we avoid computationally expensive full-depth alignment. Instead, we select a concise subset of *key layers* $\mathcal{L}_{\text{key}}$ to efficiently capture how representations evolve across layers. To address the depth mismatch between the Teacher ($N_{\mathcal{T}}$) and Student ($N_{\mathcal{S}}$), we map each selected Student layer $l_{\mathcal{S}}$ to its corresponding Teacher layer $l_{\mathcal{T}}$ via proportional scaling: $l_{\mathcal{T}} = \left\lfloor l_{\mathcal{S}} \times \frac{N_{\mathcal{T}}}{N_{\mathcal{S}}} \right\rfloor$. We define $\mathcal{L}_{\text{key}}$ using a strided top-down approach to prioritize critical high-level semantic transitions near the output. Given a stride k and a layer budget M, the selected Student indices are: $\mathcal{I}_{\mathcal{S}} = \{N_{\mathcal{S}}, N_{\mathcal{S}} - k, \ldots\}$ with $|\mathcal{I}_{\mathcal{S}}| = M$. This dis-

Method	**GPT2-120M**		**Qwen1.5-0.5B**		**OPT-1.3B**	
	λ_{DSA}	λ_{Hid}	λ_{DSA}	λ_{Hid}	λ_{DSA}	λ_{Hid}
FDD + Ours	2	0.2	2	0.2	2	0.2
DistiLLM + Ours	2	0.2	2	0.2	3	0.3
DistiLLM-2 + Ours	2	0.2	2	0.2	3	0.3

Table 10: Loss weighting coefficients used for different methods and model sizes.

cretization (typically $k \in \{2, 3\}$) effectively filters out redundant intermediate states while maintaining a coherent evolutionary trajectory. Note that while M = 3 is selected as the optimal budget for the GPT-2 pair (Section 5.3), deeper architectures such as Qwen1.5-0.5B and OPT-1.3B use a larger budget (M = 5–6) to provide sufficient supervision points along their longer representational trajectories. The budget M scales naturally with model depth via the strided top-down rule.

Model	Layer Distillation	
	Word Level	Phrase Level
GPT-2 120M	6	9, 12
Qwen1.5 0.5B	14	16, 18, 20, 22, 24
OPT 1.3B	16	18, 20, 22, 24

Table 11: Selected intermediate layers for word-level and phrase-level distillation across different models.

E Additional Ablation Results

Importance of Span Weights. To validate the necessity of our weighting mechanism, we compare our aggregation strategy against a baseline using uniform mean pooling (denoted as *w/o weight*). As shown in Table 12, removing the importance weights leads to a consistent performance degradation across both distillation frameworks.

Specifically, for the DistiLLM backbone, incorporating the weighting mechanism boosts the average ROUGE-L score from 20.66 to 21.45. Notably, performance on the *Super-Natural Instructions (S-NI)* benchmark sees a substantial improvement (+2.97 points). This indicates that not all tokens within a span contribute equally to its semantic representation. By leveraging the teacher's attention patterns to suppress non-informative tokens (e.g., stopwords or padding) and highlight salient ones, our method allows the student to capture a more precise, focused, and semantically consistent representational trajectory.

Methods	Dolly	SInst	Vicuna	S-NI	Avg.
GPT-2 1.5B → GPT-2 120M					
FDD + Ours *w/o*	25.09	12.44	17.03	25.17	19.93
FDD + Ours *w/*	25.64	13.60	17.00	25.75	20.50
DistiLLM + Ours *w/o*	25.95	14.10	16.38	26.21	20.66
DistiLLM + Ours *w/*	25.77	14.19	16.67	29.18	21.45

Table 12: Ablations on span weight mechanisms. *w/* and *w/o* denote with and without importance weighting.

F Prompt for evaluation via GPT-4

[System]
- Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to a query taken from the test dataset displayed below.
- A ground truth answer is provided and should be treated as the correct reference.
- Assess whether the assistant's response is accurate compared to the ground truth and whether the wording and explanation are appropriate and coherent.
- Begin your evaluation by providing a short explanation. Be as objective as possible.
- After providing your explanation, please rate the response on a scale of 1 to 100 by strictly following this format: "[[rating]]", for example: "Rating: [[90]]".

[Question]
{question}

[Ground truth answer]
{ground truth answer}

[The Start of Assistant's Answer]
{assistant response}
[The End of Assistant's Answer]

Figure 7: Prompt for GPT-4 evaluation using Ground Truth.