Beyond Logits: Aligning Feature Dynamics for Effective Knowledge Distillation

Guoqiang Gong[1,*], **Jiaxing Wang**[1,*], **Jin Xu**[2], **Deping Xiang**[1], **Zicheng Zhang**[1], **Leqi Shen**[3], **Yifeng Zhang**[1], **Junhua Shu**[1], **Zhaolong Xing**[1], **Zhen Chen**[1], **Pengzhang Liu**[1,†], **Ke Zhang**[1]

[1] JD.com [2] University of Oxford [3] Tsinghua University

{gongguoqiang1, wangjiaxing41,liupengzhang}@jd.com

*Equal contribution. †Corresponding author.

Abstract

Knowledge distillation (KD) compresses large language models (LLMs), known as teacher models, into lightweight versions called student models, enabling efficient inference and downstream applications. However, prevailing approaches accomplish this by predominantly focusing on matching the final output distributions of student/teacher models. Drawing on the perspective that transformers can be viewed as discretizing ordinary differential equation (ODEs) on integer time steps (corresponding to layer indices), where intermediate features evolve across layers, we argue that effective KD requires aligning the entire feature dynamics between teacher and student models, which we call feature dynamics distillation (FDD). This alignment involves matching both the feature trajectory and its first-order derivative, rather than just the final states. Our approach extends the original KD objective with two additional loss terms: layer-wise feature KD, which matches discretized feature trajectory, and layer feature delta KD, which matches first-order changes in features across adjacent layers. Extensive experiments on various tasks validate the effectiveness of our distillation method.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in various natural language processing tasks (OpenAI, 2022, 2024; Qwen, 2024; DeepSeek-AI-Group, 2024). The success is, however, largely driven by scaling up the model parameters. (e.g., 175B parameters for GPT-3 (Brown et al., 2020), 70B parameters for Qwen2.5 (Qwen, 2024) and 671B for Deepseek-V3 (DeepSeek-AI-Group, 2024)). The large model size requires substantial computation and poses significant challenges in practical deployment, especially in resource-constrained environments. By transferring knowledge from a large teacher model to a smaller student model, knowledge distillation (KD) emerges as a promising solution. The rich information learned by the teacher model helps optimize the student to achieve performance levels that would be unattainable through training the student model alone.

The backbone of LLMs, the transformer architecture (Vaswani et al.), with layer-wise skip connections, can be viewed as discretized ODEs where integer time steps correspond to layer indices (Chen et al., 2018; Lu et al., 2019). The ODE states represent intermediate features that evolve across layers from initial input embeddings to final layer features, which are then projected to logits for next-token predictions. Through this lens, student models can be understood as a coarse discretization of the underlying feature dynamics. The teacher models' more accurate representation of feature dynamics can thus guide student models toward better discretizations within their representation capacity. From this perspective, we argue that effective and comprehensive knowledge distillation should align the entire feature dynamics between student and teacher models, extending beyond the conventional approach of solely matching output distributions through various divergence losses, including Kullback-Leibler divergence (Hinton et al., 2015; Muralidharan et al., 2024; Liu et al., 2024) or reverse Kullback-Leibler divergence (Gu et al., 2024; Ko et al., 2024). To this end, we propose feature dynamics distillation (FDD), which optimizes student models to match both the discretized feature trajectory and its first-order derivative (estimated through finite differences) with the teacher models. This comes down to two additional KD loss terms: the layer-wise KD loss for trajectory matching and the layer-delta KD loss for derivative matching. We demonstrate that these additional KD loss terms consistently improve distillation effectiveness, outperforming previous state-of-the-art approaches, validating the importance of feature

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 23067–23077
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

dynamics distillation.

Our ODE perspective on KD is further supported by recent empirical observations about LLMs, including findings that intermediate layers can encode richer representations (Skean et al., 2025; Geva et al., 2022; Dar et al., 2023; Elhoushi et al., 2024) and that decoding through layer contrasting can better surface factual knowledge in LLMs (Chuang et al., 2024). The former demonstrates the necessity of conducting layer-wise KD, while the latter corroborates the importance of layer-delta KD. In addition, there is a growing body of literature exploring layer-wise distillation schemes (Jiao et al., 2020; Wang et al., 2020) by matching various aspects of teacher and student models such as intermediate features and attention scores. These studies, though heuristically established and primarily focus on BERT-based models (Devlin et al., 2019), show the promise of developing KD scheme that can more comprehensively mimic the teacher.

Extensive experiments are conducted on various tasks. The results show that our proposed FDD ensures sufficient exploitation of teacher knowledge, leading to more effective and nuanced student models that demonstrate state-of-the-art performance.

2 Preliminary

2.1 Knowledge Distillation of LLMs

A transformer-based large language model (LLM) typically consists of three components: An embedding layer $f_{\text{embed}}(\cdot)$, a stack of L transformer blocks, and a language modeling (LM) head $f_{\text{head}}(\cdot)$. Given a tokenized one-hot encoded input sequence $\mathbf{x}_{1:N} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, the embedding function $f_{\text{embed}}(\cdot)$ maps it into dense feature representations $\mathbf{h}_{1:N}(0)$. These features are then processed sequentially through L transformer layers, producing intermediate features $\{\mathbf{h}_{1:N}(1), \ldots, \mathbf{h}_{1:N}(L))\}$. Finally, the LM head $f_{\text{head}}(\cdot)$, typically a linear soft-max layer, projects the final features $\mathbf{h}_{1:N}(L)$ to vocabulary distribution for next-token predictions.

Larger LLMs typically achieve better performance but are computationally expensive during inference. Knowledge distillation (KD) is a model compression technique where the "knowledge" learned by a large, complex model (teacher $\mathcal{T}$) is transferred to a smaller, more efficient model (student $\mathcal{S}$) (Hinton et al., 2015). This can be achieved by minimizing the Kullback-Leibler divergence (KLD) between their output distributions:

$$\begin{aligned}\mathcal{L}_{\text{KD}} &= \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_{\text{KL}}(p^{\mathcal{T}}(\mathbf{x}_{i+1}|\mathbf{x}_{\leq i}) \parallel p^{\mathcal{S}}_{\mathbf{w}}(\mathbf{x}_{i+1}|\mathbf{x}_{\leq i})) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_{\text{KL}}(f^{\mathcal{T}}_{\text{head}}(\mathbf{h}^{\mathcal{T}}_i(L))) \parallel f^{\mathcal{S}}_{\text{head}}(\mathbf{h}^{\mathcal{S}}_i(L)))\end{aligned} \tag{1}$$

where $\mathbf{w}$ denotes parameters of the student model to be optimized.

2.2 Transformers as ODEs

Neural network with layer-wise residual connections (He et al., 2016) can be interpreted as discretizing an ordinary differential equation (ODE) at integer time steps (Chen et al., 2018), where the time variable corresponds to layer indices. Transformer (Vaswani et al.) also incorporates residual connections and stacks layers consisting of one multi-head attention (MHA) module and a multilayer perceptron (MLP) module, mapping from sequences to sequences:

$$\begin{aligned}\mathbf{h}_{1:N} &\leftarrow \mathbf{h}_{1:N} + \text{MHA}(\text{LN}(\mathbf{h}_{1:N})) \\ \mathbf{h}_{1:N} &\leftarrow \mathbf{h}_{1:N} + \text{MLP}(\text{LN}(\mathbf{h}_{1:N})),\end{aligned}$$

where LN denotes layer normalization (Ba et al., 2016). Therefore, it can be seen as an ODE with intermediate feature sequences as states:

$$\begin{aligned}\frac{\mathrm{d}\mathbf{h}_{1:N}(t)}{\mathrm{d}t} &= f_{\text{ode}}(\mathbf{h}_{1:N}(t); t, \mathbf{w}(t)) \\ &= \text{MHA}^{(t)}(\text{LN}^{(t)}(\mathbf{h}(t)), \mathbf{w}_{\text{MHA}}(t)) \\ &+ \text{MLP}^{(t)}(\text{LN}^{(t)}(\mathbf{h}(t)), \mathbf{w}_{\text{MLP}}(t))\end{aligned} \tag{2}$$

where $\mathbf{w}(t) = [\mathbf{w}_{\text{MHA}}(t), \mathbf{w}_{\text{MLP}}(t)]$ denotes time (layer) dependent parameters. To recover the standard transformer architecture and accommodate for alternating MHAs and MLPs, one must follow the Lie-Trotter splitting scheme (Trotter, 1959) for discretization. This ODE fully characterizes the dynamics of feature evolvement in the model forward pass (Lu et al., 2019).

3 Method

Existing knowledge distillation approaches predominantly focus on output-layer distillation. Here we argue that KD can be made more effective by aligning the entire feature dynamics between the teacher and student models, incorporating both the feature trajectory and its first-order derivative. The approach, derived from the ODE perspective, is

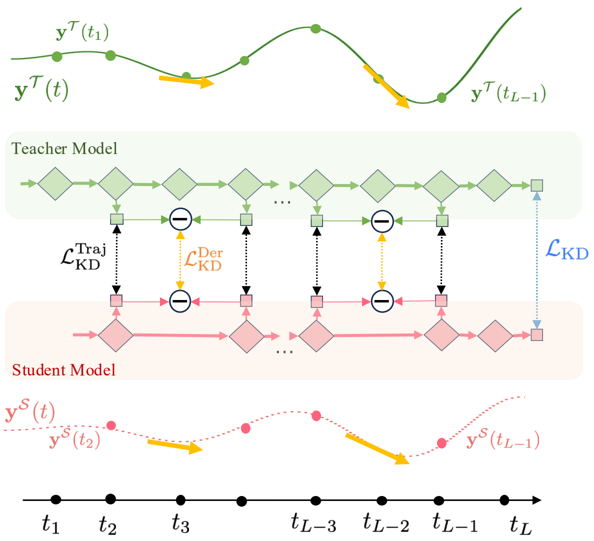
$\mathbf{y}^{\mathcal{T}}(t_1)$
$\mathbf{y}^{\mathcal{T}}(t)$
$\mathbf{y}^{\mathcal{T}}(t_{L-1})$
Teacher Model
...
$\mathcal{L}_{\text{KD}}^{\text{Traj}}$
$\mathcal{L}_{\text{KD}}^{\text{Der}}$
$\mathcal{L}_{\text{KD}}$
...
Student Model
$\mathbf{y}^{\mathcal{S}}(t)$
$\mathbf{y}^{\mathcal{S}}(t_2)$
$\mathbf{y}^{\mathcal{S}}(t_{L-1})$
t_1
t_2
t_3
t_{L-3}
t_{L-2}
t_{L-1}
t_L

Figure 1: **An illustration of feature dynamics distillation (FDD) from an ODE perspective**. The student model (in red), which represents a coarser discretization of the underlying feature dynamics ODE, is optimized to match the teacher model (in green), which is more accurate. As the feature dimensions are distinct, we map features to intermediate predictive distributions through the LM heads (green and red small rectangles) and conduct matching directly in this space. Both feature trajectory (dots) and its derivative (arrows) are matched, which comes down to a layer-wise KD loss term and a layer-delta KD loss term. The final output distribution KD loss is also incorporated.

concretized in Section 3.1. It then translates into incorporating layer-wise KD (Section 3.2) and layer-delta KD (Section 3.3) after discretization. The framework of feature dynamics distillation (FDD) is illustrated in Figure 1.

3.1 Distillation under the ODE Perspective

Given the underlying corresponding ODEs (Equation 2) for the teacher $\mathcal{T}$ and the student $\mathcal{S}$, our goal is to align their feature dynamics. Nevertheless, features in the teacher and student typically reside in different spaces with different dimensions. To bridge the gap and thus facilitate the alignment, we project both teacher and student features into the shared vocabulary prediction distribution space by multiplexing the language modeling head $f_{\text{head}}(\cdot)$ of their own. The overall dynamic is then [1]:

$$\begin{aligned}\frac{\mathrm{d}\mathbf{h}_{1:N}^{\mathcal{S}\text{ or }\mathcal{T}}(t)}{\mathrm{d}t} &= f_{\text{ode}}^{\mathcal{S}\text{ or }\mathcal{T}}(\mathbf{h}_{1:N}^{\mathcal{S}\text{ or }\mathcal{T}}(t);t)\\ \mathbf{y}_{1:N}^{\mathcal{T}\text{ or }\mathcal{S}}(t) &= \log f_{\text{head}}^{\mathcal{T}\text{ or }\mathcal{S}}(\mathbf{h}_{1:N}^{\mathcal{T}\text{ or }\mathcal{S}}(t))\end{aligned} \tag{3}$$

The prediction dynamics $\mathbf{y}(t)$ effectively capture knowledge contained in the feature dynamics $\mathbf{h}(t)$. In the following, we abuse "feature dynamics" as this "prediction dynamics". Actually, discretization of its exponential $e^{\mathbf{y}(t)}$ exactly corresponds to the Logits Lens (nostalgebraist, 2020), which has been widely applied to interpret the internals of LLM.

With the dynamics established, we align teacher-student prediction dynamics $\mathbf{y}^{\mathcal{T}}(t)$ and $\mathbf{y}^{\mathcal{S}}(t)$ from $t = 0$ to T with the trajectory loss:

$$\mathcal{L}_{\text{KD}}^{\text{Traj}} = \frac{1}{TN}\int\sum_{i=1}^{N}\mathcal{D}_{\text{KL}}(e^{\mathbf{y}_i^{\mathcal{T}}(t)} \parallel e^{\mathbf{y}_i^{\mathcal{S}}(t)})\mathrm{d}t \tag{4}$$

Matching feature trajectories alone can be insufficient due to discretization and matching errors, making it advantageous to also align the temporal evolution of features. We thus propose incorporating an additional loss term to align the first derivative of the feature trajectory:

$$\mathcal{L}_{\text{KD}}^{\text{Der}} = \frac{1}{TN}\int\sum_{i=1}^{N}\mathcal{D}_{\text{Cos}}(\frac{\mathrm{d}\mathbf{y}_i^{\mathcal{T}}(t)}{\mathrm{d}t} \parallel \frac{\mathrm{d}\mathbf{y}_i^{\mathcal{S}}(t)}{\mathrm{d}t})\mathrm{d}t \tag{5}$$

where $\mathcal{D}_{\text{Cos}}(\mathbf{a},\mathbf{b}) = 1 - \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$. The feature dynamics become very uncertain at times far from the discretization sampling points. The trajectory derivative thus provides necessary supplementary information characterizing the underline feature dynamics.

These additional knowledge distillation loss terms, $\mathcal{L}_{\text{KD}}^{\text{Traj}}$ and $\mathcal{L}_{\text{KD}}^{\text{Der}}$ derived from the ODE perspective, provide much richer training signals compared to the conventional loss based solely on output distributions.

3.2 Layer-wise Knowledge Distillation

For practical use, we discretize the ODEs Equation 3, where layer indices correspond to integer time steps. This turns out to be a layer-wise distillation scheme. As the layer number of teacher $L^{\mathcal{T}}$ and student $L^{\mathcal{S}}$ are different, we choose a distillation schedule. Teacher intermediate layers indexed by $\mathcal{I}^{\mathcal{T}} = \left\{l_{\mathcal{I}_1}^{\mathcal{T}}, \ldots, l_{\mathcal{I}_{L_D}}^{\mathcal{T}}\right\}$ are distilled to the student layers indexed by $\mathcal{I}^{\mathcal{S}} = \left\{l_{\mathcal{I}_1}^{\mathcal{S}}, \ldots, l_{\mathcal{I}_{L_D}}^{\mathcal{S}}\right\}$ correspondingly, where L_D is the number of intermediate layers to be distilled. Typically $l_{\mathcal{I}_j}^{\mathcal{T}} = K \cdot l_{\mathcal{I}_j}^{\mathcal{S}}$,

[1] Here we omit the dependence of $f_{\text{ode}}(\cdot)$ on time variant parameters $\mathbf{w}(t)$ and keep only dependence on time t for brevity.

where $K = \frac{L^{\mathcal{T}}}{L^{\mathcal{S}}}$ [2]. The feature trajectory loss Equation 4 then becomes:

$$\mathcal{L}_{\text{KD}}^{\text{Traj}} = \frac{1}{LN} \sum_{j=1}^{L_D} \sum_{i=1}^{N} \mathcal{D}_{\text{KL}}(e^{\mathbf{y}_i^{\mathcal{T}}(l_{\mathcal{I}_j}^{\mathcal{T}})} \parallel e^{\mathbf{y}_i^{\mathcal{S}}(l_{\mathcal{I}_j}^{\mathcal{S}})}) \tag{6}$$

As intermediate layers encode equal or even richer representations than those of the final layer (Skean et al., 2025), such a layer-wise distillation scheme facilitates capturing rich semantic knowledge contained in the distinct layers so that the teacher model is better exploited.

Previous layer-wise distillation methods match the teacher and student features by minimizing their L2 distance: $\frac{1}{L_D N} \sum_{j=1}^{L_D} \sum_{i=1}^{N} \|\mathbf{h}_i^{\mathcal{T}}(l_{\mathcal{I}_j}^{\mathcal{T}}) - \mathbf{P}\mathbf{h}_i^{\mathcal{S}}(l_{\mathcal{I}_j}^{\mathcal{S}})\|_2$. Generally the feature dimension $\mathrm{d}_h^{\mathcal{T}}$ and $\mathrm{d}_h^{\mathcal{S}}$ are different so the projection matrix $\mathbf{P}$ is incorporated to project the teacher and student features into the same space. In FDD, we utilize the already trained language model heads $f_{\text{head}}(\cdot)$ instead. This treatment bears a strong resemblance to the task-aware filter proposed in (Liang et al., 2023). Projection matrix trained with L2 loss indiscriminately matches the features of teacher and student. While LM head, pretrained for predicting the next token, is much more relevant to the downstream tasks(e.g. instruction tuning). As a result, it serves as a better filter for selecting knowledge that is useful for the target tasks from the hidden representations (features).

3.3 Layer-delta Knowledge Distillation

While matching the feature trajectory (layer-wise distillation) extracts rich information encoded in each layer, it does not capture the temporal evolution of features across layers due to discretization and matching errors. The feature derivative loss Equation 5 is thus incorporated. Again, with proper discretization of the ODEs, the feature derivative loss turns into a layer-delta knowledge distillation scheme:

$$\mathcal{L}_{\text{KD}}^{\text{Der}} = \frac{1}{NL_{\Delta}} \sum_{j=1}^{L_{\Delta}} \sum_{i=1}^{N} \mathcal{D}_{\text{Cos}}(\Delta_i^{\mathcal{T}}(l_{\mathcal{I}_j}^{\mathcal{T}}) \parallel \Delta_i^{\mathcal{S}}(l_{\mathcal{I}_j}^{\mathcal{S}})) \tag{7}$$

Where $L_{\Delta} = L_D - 1$. The feature changes (Δ) between adjacent selected layers are:

$$\Delta_i^{\mathcal{T}}(l_{\mathcal{I}_j}^{\mathcal{T}}) = \mathbf{y}_i^{\mathcal{T}}(l_{\mathcal{I}_j}^{\mathcal{T}}) - \mathbf{y}_i^{\mathcal{T}}(l_{\mathcal{I}_{j-1}}^{\mathcal{T}})$$
$$\Delta_i^{\mathcal{S}}(l_{\mathcal{I}_j}^{\mathcal{S}}) = \mathbf{y}_i^{\mathcal{S}}(l_{\mathcal{I}_j}^{\mathcal{S}}) - \mathbf{y}_i^{\mathcal{S}}(l_{\mathcal{I}_{j-1}}^{\mathcal{S}})$$

This layer-delta distillation loss measures the similarity of feature transitions between the teacher and the student, ensuring that the student model learns not only the knowledge encoded in each layer but also the pattern of representation refinement across layers.

3.4 Overall Optimization

The final distillation objective combines the feature trajectory loss $\mathcal{L}_{\text{KD}}^{\text{Traj}}$, the feature derivative loss $\mathcal{L}_{\text{KD}}^{\text{Der}}$ as well as the conventional KL divergence aligning the output distribution of teacher and student.

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{KD}} + \alpha \mathcal{L}_{\text{KD}}^{\text{Traj}} + \beta \mathcal{L}_{\text{KD}}^{\text{Der}} \tag{8}$$

where α, β are hyperparameters controlling the relative contributions of each loss. This comprehensive objective ensures the alignment of the teacher and the student on both semantic representations and their evolutionary patterns across layers. Ensuring a more effective student model. Note that although we utilize the conventional Kullback-Leibler divergence to establish FDD, it is compatible to other recently developed KD losses like Jensen-Shannon divergence (Agarwal et al., 2024), the reverse Kullback-Leibler divergence (Gu et al., 2024) and the Skew KLD (Ko et al., 2024).

An overall procedure of FDD is given in Algorithm 1.

Algorithm 1 Feature Dynamic Distillation (FDD)

```
Require:
    Teacher model $\mathcal{T}$; Student model $\mathcal{S}$ parameterized by $\mathbf{w}$.
    Learning rate $\eta$; Hyper-parameters $\alpha$, $\beta$.
    Selected teacher layer indexes $\mathcal{I}^{\mathcal{T}}$ and Student layer indexes $\mathcal{I}^{\mathcal{S}}$ for distillation.
Ensure:
    Distilled student model.
while not done do
    Forwarding both $\mathcal{T}$ and $\mathcal{S}$, compute the KD loss following Equation 1.
    Apply LM heads on the selected layers. Compute the feature trajectory loss $\mathcal{L}_{\text{KD}}^{\text{Traj}}$ following Equation 6.
    Compute the feature derivative loss $\mathcal{L}_{\text{KD}}^{\text{Der}}$ following Equation 7.
    Update $\mathbf{w}$ via stochastic gradient descent:
        $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \left\{ \mathcal{L}_{\text{KD}} + \alpha \mathcal{L}_{\text{KD}}^{\text{Traj}} + \beta \mathcal{L}_{\text{KD}}^{\text{Der}} \right\}$.
end while
```

[2] We can also use other distillation schemes. Different schemes can be accomplished by adjusting the teacher and student discretize step.

4 Experiments

In this section, we first compare FDD to state-of-the-art algorithms in Section 4.2. Then we analyze the effectiveness of each design ingredient in Section 4.3.

4.1 Experimental Setup

Datasets. To evaluate the effectiveness of our method, we conduct experiments on seven instruction-following benchmarks:

- **Dolly Evaluation** is a sampled test set of the `databricks-dolly-15k` dataset[3], consisting of 500 samples.
- **Self-Instruct** (Wang et al., 2023) is a user-oriented instruction-following dataset containing 252 samples.
- **Vicuna Evaluation** (Chiang et al., 2023) comprises 80 challenging questions designed to evaluate the Vicuna model.
- **Wizard Evaluation** (Xu et al., 2024) contains 218 real-world human instructions of varying difficulty across different domains from diverse sources.
- **Koala** (Geng et al., 2023) consists of 180 real user queries that span various topics and represent real-world chat system use cases.
- **Super-Natural Instructions (S-NI)** (Wang et al., 2022) includes 1,616 diverse NLP tasks with expert-written instructions, covering 76 distinct task types. The test set includes 9,000 samples spanning 119 tasks.
- **Unnatural Instructions (UnNI)** (Honovich et al., 2023) includes 240K AI-generated instructions with minimal human involvement. Following previous works (Gu et al., 2024; Ko et al., 2024), we randomly sampled 10,000 samples from the core set for evaluation.

Evaluation Metrics. Following previous methods (Gu et al., 2024; Ko et al., 2024), we utilize ROUGE-L (Lin, 2004) and GPT-4o feedback (Zheng et al., 2023) to assess the quality of model-generated outputs. ROUGE-L is a widely used evaluation metric that measures the longest common subsequence between the generated and reference texts. For the other metric, GPT-4o is tasked with comparing the model's responses to the reference answers and assigning scores ranging from 1 to 10 for each response. We calculate and report the ratio of the total scores between the model's responses and the reference answers. For the evaluation, we sample the responses using 5 random seeds and report the average scores.

Base Models. Our FDD is evaluated on three types of models with various sizes: LLaMA2 (Touvron et al., 2023) (13B teacher, 7B student), OpenLLaMA2 (Geng and Liu, 2023) (7B teacher, 3B student) and GPT2 (Radford et al., 2019) (1.5B teacher, 0.1B student).

Baselines. We compare our FDD against various state-of-the-art approaches:

- **SFT** directly fine-tunes the student model on the fixed datasets.
- **KD** (Hinton et al., 2015) applies KLD on fixed datasets.
- **SeqKD** (Kim and Rush, 2016) fine-tunes the student model using teacher-generated outputs.
- **ImitKD** (Lin et al., 2020) employs KLD on student-generated outputs (SGOs).
- **GKD** (Agarwal et al., 2024) uses Jensen-Shannon Divergence (JSD) on a combination of SGOs and fixed datasets.
- **MiniLLM** (Gu et al., 2024) utilizes reverse KLD and a policy gradient method on SGOs.
- **DistiLLM** (Ko et al., 2024) applies Skew KLD and an adaptive off-policy approach to enhance efficiency in utilizing SGOs.

Implementation Details. Our implementation is based on the experimental setup outlined by (Gu et al., 2024; Ko et al., 2024). We randomly select 14,000 samples in `databricks-dolly-15k` for training while allocating 500 samples each for validation and testing. Following previous works (Gu et al., 2024; Ko et al., 2024), the student model is first fine-tuned for 3 epochs on the training set. To ensure a fair comparison with other methods, such as ImitKD (Lin et al., 2020), GKD (Agarwal et al., 2024), MiniLLM (Gu et al., 2024), and

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

Methods	Dolly		Self-Instruct		Vicuna		Wizard	Koala	S-NI	UnNI	Average
	R-L	GPT-4o	R-L	GPT-4o	R-L	GPT-4o	GPT-4o	GPT-4o	R-L	R-L	
LLAMA2-13B → LLAMA2-7B											
Teacher (SFT)	29.27	73.46	21.68	75.73	19.77	57.98	47.87	48.50	34.10	35.49	28.26
SFT	28.38	65.59	20.32	65.68	17.72	51.93	44.56	45.00	34.37	33.25	26.81
KD (Hinton et al., 2015)	28.56	67.93	19.92	63.67	18.20	53.17	43.95	45.82	31.85	31.12	25.93
SeqKD (Kim and Rush, 2016)	28.29	72.30	19.82	64.14	17.70	51.62	44.72	43.67	30.80	31.82	25.69
ImitKD (Lin et al., 2020)	27.03	65.59	21.76	65.99	17.90	43.87	41.90	42.41	31.18	31.07	25.79
GKD (Agarwal et al., 2024)	29.27	75.07	21.25	72.79	18.61	54.10	46.49	47.31	36.94	34.69	28.15
MiniLLM (Gu et al., 2024)	30.75	76.14	23.61	73.65	**20.80**	60.24	47.10	47.87	36.17	36.82	29.63
DistiLLM (Ko et al., 2024)	31.06	76.67	23.07	70.47	20.60	58.75	47.43	47.94	37.11	37.28	29.82
Ours	**32.57**	**80.17**	**24.56**	**75.11**	19.88	**62.48**	**48.42**	**48.85**	**40.43**	**43.01**	**32.09**
OpenLLAMA2-7B → OpenLLAMA2-3B											
Teacher (SFT)	27.09	59.32	18.24	57.49	17.89	42.01	39.86	40.59	31.44	33.20	25.57
SFT	24.95	54.22	17.38	50.23	14.58	38.75	33.28	35.06	28.65	27.69	22.65
KD (Hinton et al., 2015)	25.10	52.47	17.11	51.15	16.64	38.29	35.60	37.23	28.93	28.32	23.22
SeqKD (Kim and Rush, 2016)	25.22	55.68	17.30	52.39	16.06	42.48	32.84	37.44	29.26	27.90	23.14
ImitKD (Lin et al., 2020)	23.81	53.64	17.55	51.15	16.62	40.77	37.04	37.09	29.83	28.97	23.35
GKD (Agarwal et al., 2024)	26.54	60.34	20.21	57.80	19.14	47.28	39.53	39.75	35.44	31.65	26.60
MiniLLM (Gu et al., 2024)	28.69	64.13	20.42	55.95	18.93	42.01	39.41	40.10	35.46	35.11	27.72
DistiLLM (Ko et al., 2024)	28.99	63.41	19.99	53.94	**20.41**	47.13	39.75	40.61	36.60	35.27	28.25
Ours	**30.15**	**66.61**	**20.93**	**59.96**	19.63	**48.06**	**40.85**	**42.55**	**39.88**	**38.51**	**29.82**
GPT-2 XL (1.5B) → GPT-2 (0.1B)											
Teacher (SFT)	26.34	47.95	15.27	38.63	16.96	34.88	29.13	32.05	26.71	30.18	23.09
SFT	23.53	35.42	10.49	26.58	14.98	22.79	20.56	22.18	16.78	19.95	17.15
KD (Hinton et al., 2015)	22.31	35.42	10.37	25.96	15.50	23.10	20.07	22.67	16.64	19.18	16.80
SeqKD (Kim and Rush, 2016)	24.11	36.58	11.60	28.90	14.99	21.86	19.79	22.39	19.58	22.12	18.48
ImitKD (Lin et al., 2020)	22.04	33.67	10.19	27.04	14.83	22.17	20.62	22.74	18.11	21.03	17.24
GKD (Agarwal et al., 2024)	24.36	34.83	10.45	27.20	15.73	24.65	21.01	23.23	17.16	19.86	17.51
MiniLLM (Gu et al., 2024)	24.35	36.88	10.38	27.66	**16.15**	24.96	21.12	23.65	24.76	25.11	20.15
DistiLLM (Ko et al., 2024)	25.20	37.31	12.32	28.59	15.60	24.34	21.39	23.79	23.48	26.38	20.60
Ours	**25.70**	**38.19**	**12.48**	**30.75**	15.94	**26.04**	**22.06**	**24.77**	**27.62**	**29.04**	**22.16**

Table 1: Comparison of state-of-the-art knowledge distillation methods. ROUGE-L (R-L) metric (Lin, 2004) and GPT-4o feedback scores are reported. 'Average' represents the mean ROUGE-L score across the five benchmarks. The best performance is highlighted.

DistiLLM (Ko et al., 2024), we initialize the student models using the same fine-tuned checkpoint. The hyperparameters α and β are both set to 1, and a constant learning rate of 5e-4 is applied across all experiments. For GPT-2 models, we train all parameters for 20 epochs. For LLaMA2 and OpenLLaMA2 models, we adopt the LoRA (Hu et al., 2022) technique, with training conducted over 10 epochs. Following DistiLLM (Ko et al., 2024), LoRA is applied to the query and value weights with a rank of 16. After the knowledge distillation process, the student models are selected based on their ROUGE-L scores on the validation set, and these selected checkpoints are subsequently evaluated on the test set.

For feature trajectory and its first-order derivative matching, without loss of generality, we select intermediate layers uniformly from both the student and the teacher models. Taking the student as an example, let L^S denote the total number of intermediate layers and L_D the desired number of sampled layers. The selection interval is then $Q = \lfloor L^S/(L_D + 1) \rfloor$. The indices of the sampled layers are then $\mathcal{I} = \{l \times Q \mid l \in [1, L_D]\}$. FDD enables direct decoding of intermediate features into the vocabulary space using the model's pretrained LM head (nostalgebraist, 2020). However, the LM head may be less effective in processing the intermediate features because the LM head is specifically tuned to handle the final layer output representations during pretraining. Following the methodology of Tuned Lens (Belrose et al., 2023), we optionally train lightweight adapters for each selected layer before feature dynamic distillation.

Methods	Dolly	Self-Instruct	Vicuna	S-NI	UnNI	Average
LLaMA2-13B → TinyLLaMA-1B						
Teacher (SFT)	29.27	21.68	19.77	34.10	35.49	28.26
SFT	22.60	15.80	16.03	27.37	26.72	21.70
KD	22.99	16.34	16.00	28.31	26.75	22.08
SeqKD	22.43	15.89	15.73	29.17	26.74	21.99
ImitKD	20.65	16.04	15.92	26.46	25.15	20.84
GKD	23.61	18.89	17.03	32.51	29.89	24.38
MiniLLM	26.53	18.94	17.34	34.70	32.74	26.05
DistiLLM	26.72	19.05	**18.57**	35.08	32.83	26.45
Ours	**28.97**	**19.19**	18.18	**37.09**	**36.30**	**27.95**

Table 2: Comparison of state-of-the-art knowledge distillation methods. ROUGE-L metric is reported. 'Average' represents the mean ROUGE-L score across the five benchmarks. The best performance is highlighted.

4.2 Comparation with State of the Arts

Table 1 presents the comparison results between FDD and previous state of the arts. We make the following observations:

- First, FDD consistently outperforms previous KD methods, achieving the highest average ROUGE-L scores and generally the best GPT-4o scores. For instance, in the LLaMA2-13B to LLaMA2-7B distillation setting, FDD improves the average ROUGE-L score from the previous best result (29.82) to 32.09. This demonstrates FDD's superior ability to transfer knowledge from the teacher to the student model. Unlike baseline methods that focus solely on the final output distribution, FDD provides a more comprehensive way for the student model to leverage the rich information encoded in the teacher.

- Second, the results validate the scalability of FDD across different model architectures and parameter scales. When teacher model sizes range from 1.5B to 13B, FDD consistently achieves the best or near-best performance, demonstrating its adaptability to varying model sizes. For example, in the GPT-2 XL to GPT-2 distillation setting, FDD achieves an average ROUGE-L score of 22.16, outperforming DistiLLM (20.60) and MiniLLM (20.15). Notably, in some cases, the distilled student model even surpasses the performance of the teacher model, consistent with findings from prior studies (Gu et al., 2024; Ko et al., 2024).

Table 2 presents the results when there is a large gap between models, with LLaMA2-13B (Touvron et al., 2023) serving as the teacher model and TinyLLaMA-1B (Zhang et al., 2024a) as the student model. From Table 2, we see that in this large model size gap setting, FDD still consistently outperforms baselines in most cases, showing the effectiveness of matching the whole trajectory in mitigating the distribution mismatch problem.

$\mathcal{L}_{\text{KD}}$	$\mathcal{L}_{\text{KD}}^{\text{Traj}}$	$\mathcal{L}_{\text{KD}}^{\text{Der}}$	Average
✓	-	-	19.28
✓	✓	-	21.35
✓	-	✓	20.52
✓	✓	✓	22.16

Table 3: The impact of different loss functions. 'Average' denotes the average ROUGE-L score on the five test datasets.

4.3 Ablation Studies

To evaluate the effectiveness of each design component, we conduct ablation experiments under the GPT-2 (1.5B → 0.1B) setting and assess the results with the ROUGE-L metric.

Effect of the Number of Distilled Layers The results in Figure 2 demonstrate how incorporating intermediate-layer information affects the performance of the student model. When no intermediate-layer information is used (Number of Distilled Layers = 0), the average ROUGE-L score is 19.28, serving as the baseline. Performance improves as we increase the number of distilled intermediate layers, reaching a peak ROUGE-L score of 22.16 with four sampled layers. However, further increasing the number of layers leads to slight decline in performance. We conjecture this is due to over-constraining the student model, while the student struggles to mimic the teacher behavior due to their capacity gap. Additionally, our current approach uses evenly spaced integer step sizes (layer indices) to match student and teacher models, rather than employing adaptive step sizes as suggested by ODE theory.

Effect of $\mathcal{L}_{\text{KD}}^{\text{Traj}}$ and $\mathcal{L}_{\text{KD}}^{\text{Der}}$ The experimental results in Table 3 demonstrate the effectiveness of $\mathcal{L}_{\text{KD}}^{\text{Traj}}$ and $\mathcal{L}_{\text{KD}}^{\text{Der}}$ in enhancing the performance of the student model. Specifically, incorporating $\mathcal{L}_{\text{KD}}^{\text{Traj}}$ with $\mathcal{L}_{\text{KD}}$ improves the average performance from 19.28 to 21.35, highlighting the importance of aligning semantic representations between corre-

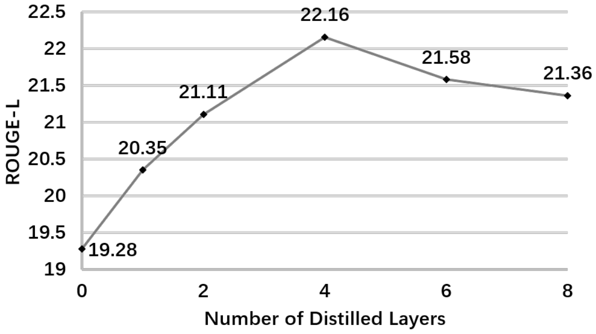

Figure 2: Effect of selecting different number of intermediate layers. Averaged ROUGE-L score on the five benchmarks are reported.

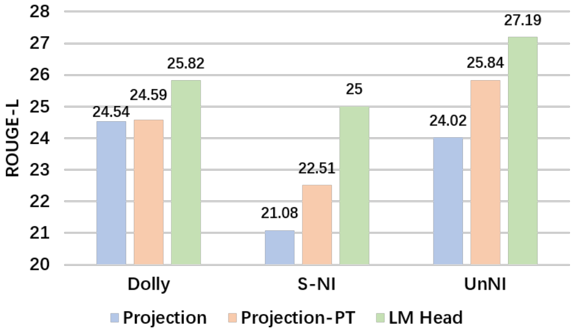

Figure 3: Ablation results of employing the LM head on the Dolly Evaluation (Dolly), Super-Natural Instructions (S-NI), and Unnatural Instructions (UnNI) datasets.

sponding intermediate layers of the teacher and student models. Similarly, adding $\mathcal{L}_{\mathrm{KD}}^{\mathrm{Der}}$ to $\mathcal{L}_{\mathrm{KD}}$ achieves an average performance of 20.52, demonstrating the value of capturing the evolution of representations across layers. Notably, the combination of $\mathcal{L}_{\mathrm{KD}}^{\mathrm{Traj}}$ and $\mathcal{L}_{\mathrm{KD}}^{\mathrm{Der}}$ with $\mathcal{L}_{\mathrm{KD}}$ achieves the highest performance of 22.16, underscoring the complementary nature of these two objectives. $\mathcal{L}_{\mathrm{KD}}^{\mathrm{Traj}}$ ensures layer-wise alignment of semantic distributions, $\mathcal{L}_{\mathrm{KD}}^{\mathrm{Der}}$ models the evolution of representations.The integration of these complementary objectives facilitates comprehensive knowledge transfer from the teacher to the student.

Effect of Multiplexing the LM Head We evaluated the effectiveness of using the LM head for intermediate layer dimension alignment between teacher and student models. As illustrated in Figure 3, we compared three alignment strategies: (1) **Projection**: This strategy randomly initializes a projection matrix as proposed in (Jiao et al., 2020). The projection matrix and student model are then jointly optimized. (2) **Projection-PT**: This approach consists of two steps. First, we freeze both teacher and student models, only the projection matrix is trained. Then, we jointly optimize the projection matrix and student model. (3) **LM Head**: This strategy directly maps intermediate layer features to the vocabulary space through the Language Model head. While Projection-PT demonstrates modest improvements over the basic Projection approach, the LM Head strategy consistently outperforms both alternative approaches across all evaluation metrics. This superior performance can be attributed to the LM Head's ability to provide stronger task-relevant signals through direct mapping of intermediate features to the vocabulary space.

5 Related Work

Knowledge Distillation of LLMs Knowledge distillation (KD) has long been an effective way to compress neural networks (Hinton et al., 2015). Standard KD minimizes the Kullback-Leibler divergence (KLD) between the student's and the teacher's output distributions (Sanh et al., 2019; Wen et al., 2023; Liu et al., 2024; Muralidharan et al., 2024; Zhong et al., 2024; Zhang et al., 2024b; Wu et al., 2025) on a fixed dataset. Recently, (Lin et al., 2020; Agarwal et al., 2024; Gu et al., 2024) argues that applying KLD incurs train-inference mismatch, thus turn to reverse Kullback-Leibler divergence (RKLD) and student-generated outputs, casting KD into a reinforcement learning problem. Orthogonal to the above works, builds upon an ODE perspective, our FDD focuses mainly on matching different quantities to fully exploit the teacher. Along this line of research, hidden states (features) (Jiao et al., 2020) and attention scores (Wang et al., 2020) have been considered. However, these approaches are only successful on BERT (Devlin et al., 2019) models. TED (Liang et al., 2023) employs task-aware filters to align hidden representations between student and teacher models at each layer. While task-aware filters function similarly to LM head, they differ in that the LM head utilizes pre-trained parameters. Besides, our FDD goes further beyond layerwise distillation. From an ODE perspective, FDD aligns the entire feature dynamics between student and teacher models, including the discretized feature trajectories (layer-wise features) as well as its first-order derivative. The effectiveness of the proposed method is validated on modern GPT-like LLMs.

Another line of research is black-box KD, also known as data KD. In practice, cases are that teacher output distribution is not accessible e.g. ChatGPT (OpenAI, 2022) APIs so the teacher is utilized as data generators (Taori et al., 2023; Peng et al., 2023; Hsieh et al., 2023; Xu et al., 2024), the resulting generated data are used for supervised fine-tuning smaller LMs. Black-box KD compromises the effectiveness of the student for teacher utility. In this paper, we consider only the white-box distillation where the teacher model is fully exploitable.

Neural ODEs Neural networks with layer-wise residual connections (He et al., 2016) have been found corresponding to Euler discretization of solutions to ordinary differential equations (ODEs) (Chen et al., 2018), with layer indices being integer time steps. Specifically, following the Lie-Trotter splitting scheme (Trotter, 1959) to accommodate for alternating MHA and MLP blocks, (Lu et al., 2019) cast the transformer (Vaswani et al.) into a convection-diffusion equation. Various novel deep architectures are proposed then from an ODE perspective, including Macaron Net (Lu et al., 2018), TransEvolve (Dutta et al., 2021), ODETransformer (Li et al., 2022) etc. Different from the above works aiming at improving the model architecture, we develop a new knowledge distillation approach from an ODE perspective to fully exploit the knowledge encoded in the teacher.

6 Conclusion

In this paper, we proposed feature dynamics distillation, a general knowledge distillation (KD) method for LLMs, inspired by viewing transformers through an ODE perspective. FDD extends beyond traditional KD methods that only match output logits by considering the entire feature dynamics, including both the discretized feature trajectory and its first-order derivative estimated through finite differences. This approach enables student models to leverage the rich information encoded in intermediate layers while mimicking the pattern of representation refinement across layers. Through extensive experiments and analysis, we demonstrated the effectiveness of this strategy.

7 Limitations

The limitations of this paper are as follows: (1) Our proposed FDD method utilizes the LM head to project intermediate layer representations of both the teacher and student models into a shared vocabulary space. Consequently, its applicability may be restricted when applied across different model architectures or vocabulary configurations. (2) Due to limited computational resources, our experiments were conducted with a teacher model of up to 13 billion parameters and a student model of up to 7 billion parameters. While the experimental results demonstrate that our method significantly outperforms existing methods at these model scales, the limitation in model size prevents us from verifying the generalizability of our findings to extremely large models, such as those with 405 billion parameters.

References

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

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. In *1607.06450*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Mateusz Litwin Eric Sigler, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2412.19437*.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 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.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Association for Computational Linguistics*.

DeepSeek-AI-Group. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Subhabrata Dutta, Tanya Gautam, Soumen Chakrabarti, and Tanmoy Chakraborty. 2021. Redesigning the transformer architecture with insights from multi-particle dynamical systems. In *Advances in Neural Information Processing Systems*.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Association for Computational Linguistics*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Conference on Empirical Methods in Natural Language Processing*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*.

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

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Annual Meeting of the Association for Computational Linguistics*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tiny{bert}: Distilling {bert} for natural language understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Conference on Empirical Methods in Natural Language Processing*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*.

Bei Li, Quan Du, Tao Zhou, Shuhan Zhou, Xin Zeng, 2 Tong Xiao1, and Jingbo Zhu. 2022. Ode transformer: An ordinary differential equation-inspired model for neural machine translation. In *Association for Computational Linguistics*.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *Conference on Empirical Methods in Natural Language Processing*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Jiaheng Liu, Chenchen Zhang, Jinyang Guo, Yuanxing Zhang, Haoran Que, Ken Deng, Zhiqi Bai, Jie Liu, Ge Zhang, Jiakai Wang, Yanan Wu, Congnan Liu, Wenbo Su, Jiamang Wang, Lin Qu, and Bo Zheng. 2024. Ddk: Distilling domain knowledge for efficient large language models. In *Advances in Neural Information Processing Systems*.

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*.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems*.

nostalgebraist. 2020. Interpreting gpt: the logit lens.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2412.19437*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Qwen. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Workshop on Energy Efficient Machine Learning and Cognitive Computing - Advances in Neural Information Processing Systems*.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

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

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

Hale F Trotter. 1959. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*.

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. In *Advances in Neural Information Processing Systems*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Annual Meeting of the Association for Computational Linguistics*.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. 2025. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *International Conference on Computational Linguistics*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024b. Dual-space knowledge distillation for large language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. In *Association for Computational Linguistics*.