AMiD: Knowledge Distillation for LLMs with α-mixture Assistant Distribution

Anonymous authors
Paper under double-blind review

Abstract

Autoregressive large language models (LLMs) have achieved remarkable improvement across many tasks but incur high computational and memory costs. Knowledge distillation (KD) mitigates this issue by transferring knowledge from a large teacher to a smaller student through distributional alignment. Previous studies have proposed various discrepancy metrics, but the capacity gap and training instability caused by near-zero probabilities, stemming from the high-dimensional output of LLMs, remain fundamental limitations. To overcome these challenges, several approaches implicitly or explicitly incorporating assistant distribution have recently been proposed. However, the past proposals of assistant distributions have been a fragmented approach without a systematic investigation of the interpolation path and the divergence. This paper proposes α-mixture assistant distribution, a novel generalized family of assistant distributions, and α-mixture distillation, coined AMiD, a unified framework for KD using the assistant distribution. The α-mixture assistant distribution provides a continuous extension of the assistant distribution by introducing a new distribution design variable α, which has been fixed in all previous approaches. Furthermore, AMiD generalizes the family of divergences used with the assistant distributions based on optimality, which has also been restricted in previous works. Through extensive experiments, we demonstrate that AMiD offers superior performance and training stability by leveraging a broader and theoretically grounded assistant distribution space.

1 Introduction

Autoregressive large language models (LLMs) have recently achieved remarkable advances, delivering outstanding performance across a wide spectrum of tasks and application domains (Achiam et al., 2023; Touvron et al., 2023; Team et al., 2024). However, their massive parameter scales impose prohibitive computational and memory costs, which hinder their deployment in practical applications. Accordingly, an essential objective for practical deployment is to compress these high-capacity models by reducing the parameter count while preserving their strong performance.

Knowledge distillation (KD) (Hinton et al., 2015) is a widely adopted compression technique that transfers knowledge from a large teacher model to a smaller student model by aligning their token-level predictive distributions. The selection of a discrepancy metric is an important research topic in KD for LLMs. Several prior studies have proposed either (1) the use of various forms of divergence, including the capability of regulating the quality-diversity trade-off (Wang et al., 2025), or (2) employing a combination of these divergences (Agarwal et al., 2024; Wu et al., 2024) as the discrepancy metric. However, these approaches do not fundamentally resolve the large capacity gap between the high-capacity teacher and smaller student models, and the optimization instability due to near-zero probabilities, which is prevalent in the high-dimensional probability space of LLMs.

A practical remedy is to introduce an *assistant distribution* that interpolates teacher and student distributions to stabilize optimization and bridge this capacity gap. Recently, several methodologies have been proposed that either (1) utilize the discrepancy metric that inherently includes a specific form of assistant distribution (Agarwal et al., 2024; Ko et al., 2024; 2025) or (2) explicitly model the assistant distribution (Shing et al., 2025). However, these approaches have generally been treated as independent recipes in different papers without a systematic study, which hinders the development of general and effective methodologies.

In this paper, we propose a generalized framework that integrates the fragmentarily employed assistant distribution and divergence. First, we interpret the existing assistant distributions from the information theory view, revealing that the existing methodology can be expressed as an m-mixture, which mixes two probability distributions via arithmetic mean, and an e-mixture, which mixes them via geometric mean. Next, we present a new assistant distribution family, coined α-mixture assistant distribution, by extending the mean concept via the generalized f_α mean. The α-mixture assistant distribution introduces a new design variable α for the assistant distribution, which adjusts the geometry of the interpolation path. Here, α is an independent parameter distinct from the well-utilized parameter λ, which controls the portion of interpolation. The α-mixture assistant distribution not only includes the existing assistant distributions as a special case ($\alpha = \pm 1$) but also provides several new assistant distribution that were not investigated in KD for LLMs area.

Under the concept of α-mixture assistant distribution, we investigate several properties of the α-mixture assistant distribution, which are meaningful in KD for LLMs, such as the analysis with α-divergence, controllable support via α, and continuity with respect to α. Next, we propose a new KD framework for LLMs, coined as $\underline{\alpha}$-$\underline{\text{mi}}$xture $\underline{\text{d}}$istillation (AMiD), which generalizes the optimization schemes of prior research by unifying both the assistant distribution and the divergence. AMiD aims to align the α-mixture assistant distribution and either the teacher or student. We theoretically prove the optimality of AMiD, which enables us to achieve the primary goal of KD (teacher = student) even when employing arbitrary divergence, α, and λ, under the perfect optimization assumption. Furthermore, through gradient analysis when employing f-divergence, we theoretically demonstrate that α adjusts the mode-covering and mode-seeking properties of the student distribution, with both toy experiments and real-world experiment results supporting this finding. Across various evaluation scenarios, our proposed framework AMiD consistently demonstrates superior performance compared to methodologies that do not utilize the assistant distribution and those employing limited assistant distribution.

2 Preliminary

2.1 Knowledge Distillation for Large Language Models

We denote the input prompt and output token sequences as x and y, respectively, where $y := (y_1, y_2, \dots, y_L) \in \mathcal{V}^L$ is a token sequence of length L, with each token drawn from the vocabulary set $\mathcal{V}$. Given the input x, an autoregressive large language model (LLM) outputs a next-token distribution $p(y_l|x, y_{<l})$, conditioned on both the prompt x and the previously generated tokens $y_{<l} := (y_1, y_2, \dots, y_{l-1})$. We assume access to two LLMs: a large fixed teacher model $p(y_l|x, y_{<l})$, and a smaller student model $q_\theta(y_l|x, y_{<l})$ parameterized by θ. The goal of knowledge distillation (KD) for LLMs is to transfer the knowledge of the teacher into the student. Concretely, KD for LLMs is typically formulated as aligning the next-token distributions of the teacher and student:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{l=1}^{L} D(p(y_l|x, y_{<l}), q_\theta(y_l|x, y_{<l})) \right] \tag{1}$$

where D denotes the divergence and the dataset $\mathcal{D}$ is composed of the predefined dataset (Hinton et al., 2015), or various strategies using the student-generated outputs (SGOs): on-policy (Lin et al., 2020), a mixed approach (Agarwal et al., 2024; Gu et al., 2024; Xu et al., 2024b), and an adaptive off-policy (Ko et al., 2024). For notational brevity, we omit the explicit dependence on x and y whenever it is clear from context, writing $p := p(y_l|x, y_{<l})$ and $q_\theta := q_\theta(y_l|x, y_{<l})$.

The choice of divergence D plays a pivotal role in KD for LLMs. The widely used Kullback–Leibler (KL) divergence $D_{\text{KL}}(p\|q_\theta) := \sum_k p(k) \log \frac{p(k)}{q_\theta(k)}$ in KD (Hinton et al., 2015; Kim & Rush, 2016) emphasizes mode-covering, often assigning mass to less informative regions. To mitigate this effect, the reverse KL divergence $D_{\text{RKL}}(p\|q_\theta) := D_{\text{KL}}(q_\theta\|p)$ is employed for its mode-seeking properties (Gu et al., 2024), which possesses mode-seeking properties, but either choice entails a trade-off between quality and diversity. Recent studies address this by (1) combining divergences, e.g., GKD (Agarwal et al., 2024) with the generalized Jensen–Shannon divergence $D_{\text{GJS}}(p|q_\theta) := \lambda D_{KL}(p\|\lambda p + (1-\lambda)q_\theta) + (1-\lambda) D_{KL}(q_\theta\|\lambda p + (1-\lambda)q_\theta)$, and (2) extending classical divergences to enable explicit control, as in ABKD (Wang et al., 2025), which adopts the α-β-divergence D_{AB} (Cichocki et al., 2011) as a generic framework.

Meanwhile, several methodologies have recently been proposed to improve the optimization stability of KD for LLMs. Ko et al. (2024) leverages the skew KL divergence $D_{\mathrm{SKL}}(p\|q_\theta) := D_{\mathrm{KL}}(p\|\lambda p + (1-\lambda)q_\theta)$ and the skew reverse KL divergence $D_{\mathrm{SRKL}}(p\|q_\theta) := D_{KL}(q_\theta\|\lambda p + (1-\lambda)q_\theta)$. TAID (Shing et al., 2025) introduces an adaptive intermediate distribution that gradually shifts from the student's initial distribution to the teacher distribution, i.e., $D_{\mathrm{TAID}}(p\|q_\theta) := D_{\mathrm{KL}}(r_t\|q_\theta)$ where $r_t := \mathrm{softmax}((1-\lambda_t)\cdot \mathrm{logit}(q'_\theta) + \lambda_t \cdot \mathrm{logit}(p))$ with time-dependent interpolation parameter λ_t, detached student logits $\mathrm{logit}(q'_\theta)$, and teacher logits $\mathrm{logit}(p)$.

2.2 m-MIXTURE AND e-MIXTURE

Mixture models are a standard tool for integrating information from multiple distributions. Information geometry (Amari, 2016; Nielsen, 2020; Eguchi & Komori, 2022) provides a dualistic structure on the manifold of probability distributions, characterized by two affine connections: the mixture connection and the exponential connection. These connections induce two natural ways of interpolating between distributions, commonly referred to as the m-*mixture* and the e-*mixture*.

Given two probability distributions p and q defined on the same measureable space, the m-mixture is defined as a convex combination of p and q:

$$p^{(m)}(x) := (1-t)p(x) + tq(x), \qquad t \in [0,1] \tag{2}$$

In contrast, the e-mixture is defined multiplicatively:

$$p^{(e)}(x) := \frac{p(x)^t q(x)^{1-t}}{Z(t)}, \qquad Z(t) := \int p(x)^t q(x)^{1-t} dx \tag{3}$$

The m-mixture forms a straight line in probability space, while the e-mixture forms one in log-probability space. Some studies leverage m- and e-mixtures, for example, to construct paths for annealed importance sampling (Grosse et al., 2013; Masrani et al., 2021).

2.3 GENERALIZED f-MEAN

Generalized f-mean (Kolmogorov & Castelnuovo, 1930) is a generalized framework of the mean by using a monotonically increasing differentiable function $f : \mathbb{R} \to \mathbb{R}$. Given a set of weights $\{w_i \in \mathbb{R}^+ \mid \sum_i w_i = 1\}$ and the set of corresponding input elements $\{u_i \in \mathbb{R}\}$, the generalized f-mean is defined as:

$$m_f(\{w_i\},\{u_i\}) := f^{-1}\left(\sum_i w_i f(u_i)\right) \tag{4}$$

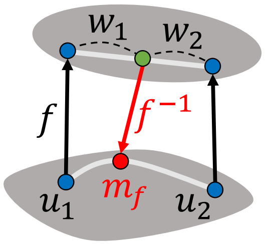


Figure 1: Illustration of generalized f-mean.

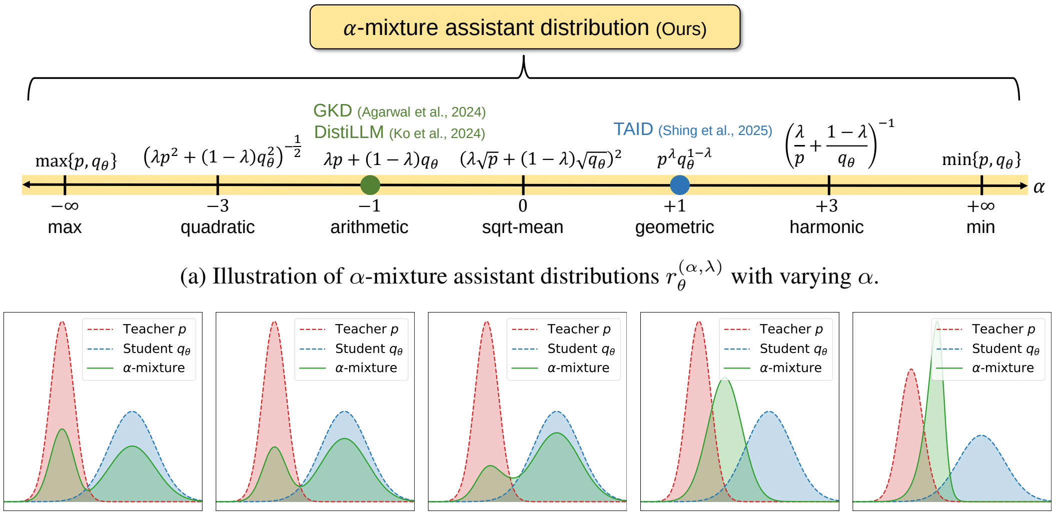
The m_f applies a nonlinear transformation to the inputs, combines them with weights in the transformed domain, and maps the result back to the original domain. The well-known means, such as the arithmetic mean and geometric mean, have *homogeneity*, which stands for a scale-free property $m_f\big(\{w_i\},\{c\cdot u_i\}\big) = c\cdot m_f\big(\{w_i\},\{u_i\}\big)$ for $c > 0$. The generalized f-mean is homogeneous only when f belongs to the unique class of functions (Hardy, 1952; Amari, 2007):

$$f(u) := f_\alpha(u) = \begin{cases} u^{\frac{1-\alpha}{2}}, \ \alpha \neq 1 \\ \log u, \ \ \alpha = 1 \end{cases}, \qquad u \in \mathbb{R}^+ \tag{5}$$

This family includes various notable examples, such as the weighted arithmetic mean for $\alpha = -1$, the weighted geometric mean for $\alpha = 1$, the weighted harmonic mean for $\alpha = 3$, and $\min\{u_i\}, \max\{u_i\}$ for $\alpha \to \infty$ and $\alpha \to -\infty$, respectively.

3 METHODOLOGY

This section introduces a new KD framework for LLMs, coined α-mixture distillation (AMiD), which generalizes both the assistant distribution and the associated optimization scheme. Section 3.1 reveals the connection among the existing assistant distributions and highlights the need for a systematic study. Section 3.2 proposes the α-mixture assistant distribution, which provides a unified and generalized assistant distribution family via α-mixture distribution. Finally, Section 3.3 extends the assistant-based KD objective into a generic divergence framework.



(a) Illustration of α-mixture assistant distributions $r_\theta^{(\alpha,\lambda)}$ with varying α.

(b) $\alpha = -3$ (c) $\alpha = -1$ (d) $\alpha = 0$ (e) $\alpha = 1$ (f) $\alpha = 3$

Figure 2: Visualization of the α-mixture assistant distribution family. (a) The α-mixture assistant distribution provides a generalized framework for assistant distributions, with prior studies (Agarwal et al., 2024; Ko et al., 2024; Shing et al., 2025) recoverable as special cases. (b-f) Illustration of the α-mixture assistant distribution where $p = \mathcal{N}(0, 0.5^2)$, $q_\theta = \mathcal{N}(3, 1^2)$, and $\lambda = 0.3$. For $\alpha < 1$, the support of the α-mixture assistant distribution corresponds to the union of the supports of p and q_θ, whereas for $\alpha \geq 1$, it corresponds to their intersection.

3.1 Motivation

Our primary motivation stems from the observation that recent studies inherently include the composition of the teacher distribution p and the student distribution q_θ, which we will refer as *assistant distribution* r_θ in this paper. For example, several studies (Agarwal et al., 2024; Ko et al., 2024) utilize the divergences that include $r_\theta := \lambda p + (1 - \lambda) q_\theta$ with $\lambda \in [0, 1]$, which is an weighted arithmetic mean, also known as m-*mixture*. Moreover, we have newly discovered that the assistant distribution of TAID (Shing et al., 2025) is e-*mixture*, also known as a weighted geometric mean.

Proposition 3.1. *The assistant distribution of TAID (Shing et al., 2025) is* e*-mixture of* p *and* q_θ*:*[1]

$$r_\theta := \mathrm{softmax}((1 - \lambda) \cdot \mathrm{logit}(q_\theta) + \lambda \cdot \mathrm{logit}(p)) \propto p^\lambda q_\theta^{1-\lambda} \tag{6}$$

Please refer to Appendix A.2 for proof. Using the assistant distribution provides several advantages in KD for LLMs. First, the assistant distribution facilitates more effective knowledge transfer between the teacher and the student. In KD, a significant capacity gap often arises due to differences in model size (Mirzadeh et al., 2020), and this issue becomes particularly pronounced in LLMs (Zhang et al., 2023; Sun et al., 2025) due to the high-dimensional nature. This gap makes it difficult for the student to faithfully capture the knowledge encoded in the teacher (Mirzadeh et al., 2020; Shing et al., 2025). By introducing the assistant distribution that serves as a bridge between the teacher and student, the information transfer might be more efficient (Shing et al., 2025). Second, the assistant distribution improves training stability. Due to the high-dimensional nature of LLMs, most of probabilites in p and q_θ are inevitably close to zero. These near-zero probabilities might cause instability in both the loss and the gradient computation when divergences involving density ratios (e.g., KL divergence) are used (Ko et al., 2024). A suitably constructed assistant distribution yields more stable density-ratio estimates, thereby enhancing the robustness of optimization (Ko et al., 2024).

Despite these advantages, no systematic study has examined (1) the distinction between m- and e-mixture assistant distributions, (2) alternative candidates, (3) their compatibility with diverse divergences, and (4) their implications for KD in LLMs, supported by theoretical and empirical analyses. This gap hinders the development of general and effective methodologies, so the recent studies often fall into sub-optimal performances by relying on an isolated design of assistant distribution. In this paper, we alleviate this gap by unifying the existing assistant distributions into a generalized design principle of assistant distribution.

[1] We omit the time index t and detached notation for the sake of uniformity.

3.2 α-MIXTURE ASSISTANT DISTRIBUTION

As discussed in Section 3.1, the existing assistant distributions can be formulated as the mixture of the teacher distribution and the student distribution via the mean function. To integrate the fragmentarily employed assistant distributions, we employ the generalized f_α-mean (Amari, 2016) and introduce a new assistant distribution family, coined *α-mixture assistant distribution* as follows:

Definition 1 (*α-mixture assistant distribution*)**.** Let $\alpha \in \mathbb{R}$ and $\lambda \in [0, 1]$. For distributions p and q_θ defined either on a discrete support $\mathcal{X}$ indexed by k or on a continuous domain $\mathcal{X}$ with variable x, define the unnormalized α-mixture assistant distribution as:

$$\tilde{r}_\theta^{(\alpha,\lambda)}(z) = \begin{cases} \left(\lambda\, p(z)^{\frac{1-\alpha}{2}} + (1-\lambda)\, q_\theta(z)^{\frac{1-\alpha}{2}} \right)^{\frac{2}{1-\alpha}}, & \text{if } \alpha \neq 1, \\ p(z)^{\lambda}\, q_\theta(z)^{1-\lambda}, & \text{if } \alpha = 1, \end{cases} \tag{7}$$

where $z = k$ in the discrete case and $z = x$ in the continuous case.

Consequently, the (normalized) α-mixture assistant distribution is defined as:

$$r_\theta^{(\alpha,\lambda)}(z) = \frac{\tilde{r}_\theta^{(\alpha,\lambda)}(z)}{Z_r}, \qquad Z_r := \sum_k \tilde{r}_\theta^{(\alpha,\lambda)}(k) \text{ or } \int_{\mathcal{X}} \tilde{r}_\theta^{(\alpha,\lambda)}(x)\, dx \tag{8}$$

The α-mixture assistant distribution $r_\theta^{(\alpha,\lambda)}$ contains two tunable parameters: α and λ. The λ determines the portion of the interpolation between teacher p and student model q_θ, which has been fine-tuned in previous works (Agarwal et al., 2024; Ko et al., 2024; Shing et al., 2025). The other parameter α is a new axis of distribution design variable, which was only employed as a specialized case ($\alpha = \pm 1$), controls the geometry of the interpolation path, as depicted in Figure 3a. Since the form of the generalized f_α-mean is solely governed by α, once α is fixed, λ only serves to control the portion between p and q_θ along that determined path. In addition, Theorem 3.2 provides an helpful information geometric perspective of the α-mixture (assistant) distribution:

Theorem 3.2. *(Amari, 2007) Given a fixed α and λ, the $r^{(\alpha,\lambda)}$ defined as Eq. (8) is unique minizer of a weighted sum of Amari's α-dviergences D_α (see Appendix A.1 for the definition):*

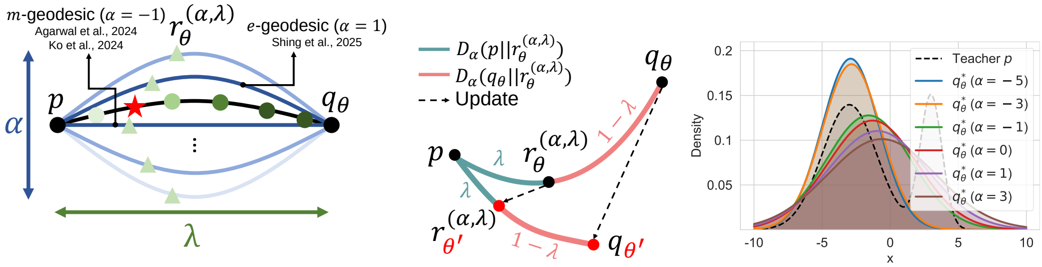
$$r^{(\alpha,\lambda)} = \arg\min_r \lambda \cdot D_\alpha(p \| r) + (1-\lambda) \cdot D_\alpha(q \| r) \tag{9}$$

Theorem 3.2 indicates that $r_\theta^{(\alpha,\lambda)}$ is the internal division distribution of p and q_θ in terms of α-divergence, which bridges the generalization of the mean concept and the geodesic in information geometry. Due to the generalized f_α-mean, the existing assistant distributions are recoverable as special instances of $r_\theta^{(\alpha,\lambda)}$: $r_\theta^{(-1,\lambda)}$ is m-mixture (Agarwal et al., 2024; Ko et al., 2024) that is minimizer of a weighted sum of D_{KL}, and $r_\theta^{(1,\lambda)}$ is e-mixture (Shing et al., 2025) that is minimizer of a weighted sum of D_{RKL}. Furthermore, $r_\theta^{(\alpha,\lambda)}$ provides several new assistant distributions that were not previously used in KD literature, as depicted in Figure 2a.

Moreover, the support of α-mixture assistant distribution determined by the range of α: $\text{supp}(r_\theta^{(\alpha,\lambda)}) = \text{supp}(p) \cup \text{supp}(q_\theta)$ when $\alpha < 1$, and $\text{supp}(r_\theta^{(\alpha,\lambda)}) = \text{supp}(p) \cap \text{supp}(q)$ when $\alpha \geq 1$. This property demonstrates the necessity of determining the range of α based on the characteristics at the intersection of p and q_θ. For instance, if p and q_θ overlap significantly, setting $\alpha \geq 1$ can strengthen the matching within the intersection region. Conversely, if they overlap minimally, setting $\alpha < 1$ indicates that matching occurs across a broader range. Although in KD for LLMs, p and q_θ typically share the same support defined by the vocabulary set, this property remains useful because many probabilities are very small values close to zero due to the high dimensionality. Figures 2b-2f shows the different behaviors of $r_\theta^{(\alpha,\lambda)}$ among the various α values.

Lastly, we also demonstrate that $r_\theta^{(\alpha,\lambda)}$ is a continuous function with respect to α in Proposition 3.3, even though the $r_\theta^{(\alpha,\lambda)}$ is a piecewise-defined function. This property enables the design of a curriculum-based adaptive α scheduling, paralleling prior work (Shing et al., 2025; Ko et al., 2025) that investigated adaptive strategies for λ. Please refer to Appendix A.3 for proof.

Proposition 3.3. *(Continuity) Assume that p and q_θ are not both zero. Then, $r_\theta^{(\alpha,\lambda)}$ is continuous function w.r.t α under the fixed $\lambda \in [0, 1]$.*



(a) Role of α and λ in the distribution space. (b) Optimization dynamics. (c) Role of α for q_θ^*.

Figure 3: Visualization of the characteristics of the α-mixture distillation, AMiD. (a) α determines the geometry of interpolation while λ controls the portion between p and q_θ. In general, the curvature of the path increases as α moves farther away from -1 (the straight line). (b) $r_\theta^{(\alpha,\lambda)}$ can be interpreted as the internal division point in terms of α-divergence. Due to the uniqueness, the updated parameter θ' via optimization of AMiD, directly affects the student distribution. (c) Toy experiment with two-modal p and uni-modal q_θ. α controls the property of the optimized student distribution q_θ^* between the mode-covering and mode-seeking, even though we minimize the fixed divergence $D_{\text{KL}}(p\|r_\theta^{(\alpha,\lambda)})$. In particular, when $\alpha \leq 1$, increasing α encourages q_θ^* to exhibit more mode-covering behavior, whereas using smaller α strengthens mode-seeking behavior.

3.3 AMiD: Knowledge distillation with α-mixture assistant distribution

In this section, we present a token-level KD for LLMs with α-mixture assistant distribution, coined as $\underline{\alpha}$-$\underline{\text{mi}}$xture $\underline{\text{d}}$istillation (AMiD), which aims to align $r_\theta^{(\alpha,\lambda)}$ and either p or q_θ. Specifically, the optimization of AMiD is defined as follows, similar to Eq. (1):

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_{l=1}^{L} D(p, r_\theta^{(\alpha,\lambda)})\right] \quad \text{or} \quad \min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_{l=1}^{L} D(q_\theta, r_\theta^{(\alpha,\lambda)})\right] \tag{10}$$

We highlight that AMiD allows the use of arbitrary divergence D and any dataset $\mathcal{D}$ (see Section 2.1) since $r_\theta^{(\alpha,\lambda)}$ is a valid distribution. Furthermore, AMiD generalizes the optimization schemes of prior research by extending both the assistant distribution and the divergence. For example, DistiLLM (Ko et al., 2024) corresponds to $D_{\text{KL}}(p\|r_\theta^{(-1,\lambda)})$ and $D_{\text{KL}}(q_\theta\|r_\theta^{(-1,\lambda)})$; and TAID (Shing et al., 2025) corresponds to $D_{KL}(r_\theta^{(1,\lambda)}\|q_\theta)$. Next, we aim to characterize the optimality of AMiD.

Theorem 3.4. *(Optimality) Let D be any proper divergence and $\alpha \in \mathbb{R}$.*

- *If $\lambda \in [0, 1)$ and $\exists\theta$ s.t. $D(p, r_\theta^{(\alpha,\lambda)}) = 0$, then $D(p, r_\theta^{(\alpha,\lambda)}) = 0$ if and only if $p = q_\theta$.*
- *If $\lambda \in (0, 1]$ and $\exists\theta$ s.t. $D(q_\theta, r_\theta^{(\alpha,\lambda)}) = 0$, then $D(q_\theta, r_\theta^{(\alpha,\lambda)}) = 0$ if and only if $p = q_\theta$.*

Please refer to Appendix B.3 for the proof. Theorem 3.4 demonstrates that even if we minimize the divergence between p (or q_θ) and $r_\theta^{(\alpha,\lambda)}$, the primary goal of KD is guaranteed i.e., $p = q_\theta$. It is intuitive because the interpolation point needs to coincide with one of the endpoints when it coincides with the other (see Figure 3b). Therefore, leveraging the benefits of the assistant distribution, we establish optimality. Although Theorem 3.4 establishes theoretical optimality for any choice of D, $\alpha \in \mathbb{R}$, and $\lambda \in (0, 1)$, the effectiveness of AMiD might depend on selecting an appropriate α value due to the imperfect practical optimization. Please refer to Appendix B.3 for the theoretical insights based α tuning guidelines, Appendix C.4 for overlap-based adaptive α scheduling.

Now, we provide the gradient analysis to investigate the specific role of α. In particular, we consider f-divergence, which is widely used in many areas, including KD for LLMs.

Proposition 3.5. *(Gradient analysis) The gradient of f-divergence $D_f(p||r_\theta^{(\alpha,\lambda)})$ be expressed as:*

$$\nabla_\theta D_f\left(p||r_\theta^{(\alpha,\lambda)}\right) = \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[w \cdot \left\{\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right) - \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right)\right]\right\} \cdot \nabla_\theta \log q_\theta\right] \tag{11}$$

where $w := \frac{(1-\lambda)q_\theta^{\frac{1-\alpha}{2}}}{\lambda p^{\frac{1-\alpha}{2}} + (1-\lambda)q_\theta^{\frac{1-\alpha}{2}}}$ and $\psi_f(v) := f(v) - vf'(v)$.

Table 1: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets. **Bold** and Underline mean the best and second-best performance of each column, except the teacher, respectively. All results are based on our own re-implementation. We use D_{AB} and $\lambda = 0.1$ for AMiD. We conduct the evaluation with five random seeds. More results of baselines are in Appendix C.1.

Model	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
GPT-2 XL (Teacher)	–	27.14 ±0.15	14.55 ±0.82	16.12 ±0.31	27.21 ±0.25	31.41 ±0.06	23.29
GPT-2 XL (1.5B) → GPT-2 (0.1B)							
GKD	27.06	24.58 ±0.13	11.78 ±0.44	14.60 ±0.37	22.84 ±0.12	25.04 ±0.09	19.77
TAID	28.37	25.74 ±0.27	12.91 ±0.31	17.09 ±0.18	23.66 ±0.31	26.82 ±0.05	21.24
DistiLLM (SKL)	27.88	25.50 ±0.28	12.35 ±0.39	16.10 ±0.22	23.87 ±0.39	26.16 ±0.06	20.80
DistiLLM (SRKL)	28.21	25.74 ±0.20	12.13 ±0.23	16.34 ±0.15	25.40 ±0.10	26.91 ±0.12	21.30
ABKD	28.61	25.49 ±0.24	12.52 ±0.52	**17.36** ±0.55	26.07 ±0.14	27.36 ±0.10	21.76
AMiD (Ours)	**29.24**	**26.44** ±0.12	**13.74** ±0.49	16.76 ±0.24	**29.71** ±0.08	**30.35** ±0.09	**23.40**
GPT-2 XL (1.5B) → GPT-2 Medium (0.3B)							
GKD	27.90	25.06 ±0.55	12.36 ±0.42	15.71 ±0.58	23.83 ±0.26	27.14 ±0.09	20.82
TAID	29.45	27.01 ±0.27	14.53 ±0.47	17.58 ±0.20	25.14 ±0.15	29.79 ±0.14	22.81
DistiLLM (SKL)	29.65	26.87 ±0.13	14.11 ±0.29	16.85 ±0.54	25.59 ±0.22	28.84 ±0.03	22.45
DistiLLM (SRKL)	29.72	26.50 ±0.20	13.79 ±0.71	17.14 ±0.52	26.25 ±0.11	29.31 ±0.16	22.60
ABKD	29.64	26.93 ±0.17	13.69 ±0.32	17.45 ±0.27	28.15 ±0.18	30.94 ±0.06	23.43
AMiD (Ours)	**30.83**	**27.34** ±0.18	**15.26** ±0.46	**17.69** ±0.27	**29.04** ±0.20	**33.15** ±0.13	**24.50**
GPT-2 XL (1.5B) → GPT-2 Large (0.8B)							
GKD	29.36	26.38 ±0.24	14.44 ±0.66	17.02 ±0.46	26.64 ±0.16	30.99 ±0.13	23.09
TAID	29.83	26.85 ±0.32	15.07 ±0.31	17.02 ±0.48	26.71 ±0.23	31.09 ±0.17	23.35
DistiLLM (SKL)	29.69	26.12 ±0.27	15.69 ±0.75	16.91 ±0.43	27.23 ±0.18	30.73 ±0.12	23.34
DistiLLM (SRKL)	30.59	27.09 ±0.40	14.61 ±0.66	16.39 ±0.27	28.44 ±0.45	31.04 ±0.06	23.51
ABKD	30.49	27.67 ±0.34	15.46 ±0.81	**17.43** ±0.25	30.74 ±0.22	33.11 ±0.15	24.88
AMiD (Ours)	**31.10**	**27.86** ±0.29	**16.46** ±0.41	16.62 ±0.50	**32.64** ±0.26	**35.64** ±0.07	**25.84**

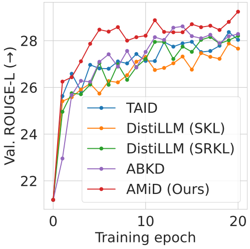


Figure 4: ROUGE-L curve on Dolly dataset.

Table 2: Experimental results on the task-specific distillation. "Trans." and "Summ." indicate translation and summarization task, respectively. We use D_{AB} and $\lambda = 0.1$ for AMiD.

Model	SFTed Gemma-7B-It → Gemma-2B-It			SFTed Qwen2-7B-It to Qwen2-0.5B-It		
	Trans. COMET (↑)	Summ. ROUGE-L (↑)	GSM8K. Acc (↑)	Trans. COMET (↑)	Summ. ROUGE-L (↑)	GSM8K Acc (↑)
TAID	74.20	34.93	24.49	58.17	31.65	33.28
DistiLLM	52.83	26.51	00.00	57.23	32.27	35.63
ABKD	74.21	34.88	24.26	58.07	31.67	33.13
AMiD ($\alpha \neq \pm 1$)	**74.78**	**35.22**	**24.94**	**58.31**	**32.51**	**36.24**

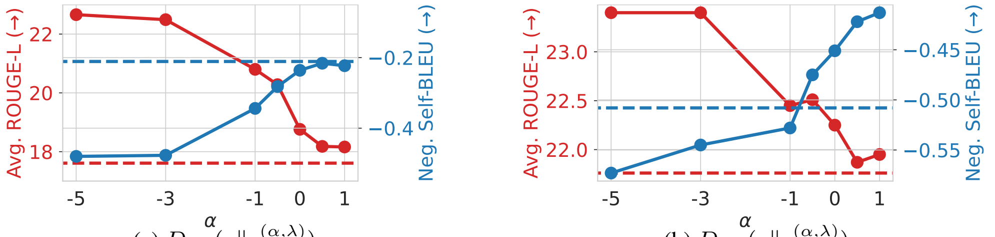
Please refer to Appendix A.5 for the proof. Proposition 3.5 implies the following properties. First, w controls the magnitude of the instance-wise gradient $\nabla_\theta \log q_\theta(y_l \mid x, y_{<l})$. While w does not affect the individual gradient direction due to $0 \leq w \leq 1$, its weighting effect may shift the batch-wise gradient direction. Second, the α plays a crucial role in enabling w to perform instance-wise weighting based on the density ratio $\frac{p(y_l|x,y_{<l})}{q_\theta(y_l|x,y_{<l})}$. This property originates from the unique characteristic of the α-mixture assistant distribution that cannot be achieved by λ or learning rate scheduling. Third, α (relatively) adjusts the *mode-covering* and *mode-seeking* behavior of the optimized student distribution. Let us assume $\frac{1-\alpha}{2} \geq 0$, i.e., $\alpha \leq 1$.[2] When $p \geq q_\theta$, a larger α produces a correspondingly larger value of w. Thus, it amplifies the gradient magnitude in regions where the student underestimates the teacher. As a result, choosing a large α (relatively) encourages the student distribution q_θ to exhibit a *mode-covering* behavior. In contrast, employing the small α in $p < q_\theta$ results in a large w value. It assigns a large gradient magnitude to the area where the student overestimates, ultimately exhibiting that the small α (relatively) reinforces *mode-seeking* property.

To support the results from the gradient analysis, we investigate the property of the optimized student distribution q_θ^* through the toy experiments. Figure 3c shows that the optimized student distribution q_θ^* with small α converges to one of the peak, which indicates the mode-seeking. As increasing α, the q_θ^* gradually have thick tails while moving towards the average of p, which implies the mode-covering. These analyses indicate that the balance between mode-covering and mode-seeking, often attributed to divergence selection, might be controlled by α in the α-mixture assistant distribution.

[2] Although α can be any real number, we assume $\alpha \leq 1$ for the sake of simplicity.

Table 3: ROUGE-L scores (↑) with various divergences D and α. We utilize GPT-2 XL (1.5B) → GPT-2 (0.1B). We use $\lambda = 0.1$ for AMiD. q_θ indicates the no-assistant baseline. $\alpha = -1$ and $\alpha = 1$ correspond to the assistant distribution employed in DistiLLM and TAID, respectively.

Divergence D	Assistant $r_\theta^{(\alpha,\lambda)}$	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
$D_{\mathrm{KL}}(p\|r_\theta^{(\alpha,\lambda)})$	q_θ	25.25	22.96 ±0.23	10.54 ±0.14	15.33 ±0.13	18.10 ±0.26	21.10 ±0.16	17.61
	AMiD ($\alpha = -5.0$)	**28.99**	**25.86** ±0.10	**13.72** ±0.42	15.90 ±0.29	**28.32** ±0.24	**29.52** ±0.06	**22.66**
	AMiD ($\alpha = -3.0$)	28.47	25.72 ±0.17	13.68 ±0.19	**16.71** ±0.30	27.30 ±0.30	29.03 ±0.12	22.49
	AMiD ($\alpha = -1.0$)	27.88	25.50 ±0.28	12.35 ±0.39	16.10 ±0.22	23.87 ±0.39	26.16 ±0.06	20.80
	AMiD ($\alpha = -0.5$)	27.37	24.17 ±0.37	12.15 ±0.49	16.37 ±0.38	24.34 ±0.20	24.36 ±0.06	20.28
	AMiD ($\alpha = 0.0$)	26.37	24.08 ±0.25	10.65 ±0.20	16.27 ±0.24	20.09 ±0.20	22.71 ±0.13	18.76
	AMiD ($\alpha = 0.5$)	25.56	22.81 ±0.22	10.77 ±0.40	16.24 ±0.23	18.96 ±0.45	22.13 ±0.10	18.18
	AMiD ($\alpha = 1.0$)	25.31	22.99 ±0.12	11.17 ±0.51	15.97 ±0.46	18.74 ±0.40	21.94 ±0.16	18.16
$D_{\mathrm{RKL}}(p\|r_\theta^{(\alpha,\lambda)})$	q_θ	28.85	**26.67** ±0.50	12.32 ±0.43	17.48 ±0.30	24.25 ±0.19	26.56 ±0.19	21.46
	AMiD ($\alpha = -5.0$)	28.39	26.16 ±0.33	**12.95** ±0.57	17.39 ±0.39	24.59 ±0.22	27.17 ±0.09	21.65
	AMiD ($\alpha = -3.0$)	28.75	26.47 ±0.12	12.71 ±0.21	17.17 ±0.42	**27.00** ±0.11	**28.16** ±0.12	**22.30**
	AMiD ($\alpha = -1.0$)	**28.97**	22.96 ±0.23	12.34 ±0.24	17.27 ±0.64	22.44 ±0.36	25.68 ±0.10	20.83
	AMiD ($\alpha = -0.5$)	28.25	26.15 ±0.30	11.81 ±0.29	16.54 ±0.22	23.49 ±0.25	25.87 ±0.06	20.77
	AMiD ($\alpha = 0.0$)	28.80	25.84 ±0.22	12.06 ±0.13	**17.71** ±0.65	22.72 ±0.24	25.36 ±0.10	20.74
	AMiD ($\alpha = 0.5$)	28.45	25.42 ±0.11	11.45 ±0.26	17.31 ±0.38	21.58 ±0.24	24.43 ±0.10	20.04
	AMiD ($\alpha = 1.0$)	0.16	4.27 ±0.01	2.81 ±0.02	9.12 ±0.06	1.64 ±0.00	1.84 ±0.0	3.94
$D_{\mathrm{AB}}(p\|r_\theta^{(\alpha,\lambda)})$	q_θ	28.61	25.49 ±0.24	12.52 ±0.52	17.36 ±0.55	26.07 ±0.14	27.36 ±0.10	21.76
	AMiD ($\alpha = -5.0$)	**29.24**	**26.44** ±0.12	**13.74** ±0.49	16.76 ±0.24	**29.71** ±0.08	**30.35** ±0.09	**23.40**
	AMiD ($\alpha = -3.0$)	29.07	26.38 ±0.18	13.58 ±0.57	16.11 ±0.18	29.27 ±0.14	30.14 ±0.06	23.10
	AMiD ($\alpha = -1.0$)	28.70	26.10 ±0.24	13.34 ±0.25	16.71 ±0.27	26.55 ±0.17	29.55 ±0.11	22.45
	AMiD ($\alpha = -0.5$)	28.70	26.37 ±0.27	13.59 ±0.25	**17.02** ±0.34	27.06 ±0.31	28.50 ±0.16	22.51
	AMiD ($\alpha = 0.0$)	28.86	25.77 ±0.34	13.57 ±0.22	16.14 ±0.36	27.26 ±0.27	28.52 ±0.20	22.25
	AMiD ($\alpha = 0.5$)	28.46	25.80 ±0.32	12.94 ±0.36	16.59 ±0.34	26.29 ±0.22	27.73 ±0.08	21.87
	AMiD ($\alpha = 1.0$)	24.93	22.36 ±0.22	9.72 ±0.58	16.29 ±0.30	15.09 ±0.19	16.15 ±0.12	15.92



(a) $D_{\mathrm{KL}}(p\|r_\theta^{(\alpha,\lambda)})$ (b) $D_{\mathrm{AB}}(p\|r_\theta^{(\alpha,\lambda)})$

Figure 5: Performance curve on ROUGE-L (quality) and Self-BLEU (diversity). Colored dashed lines: no-assistant baseline (q_θ). We use $\lambda = 0.1$.

4 Experiments

We consider both general instruction-following distillation and task-specific distillation to validate the effectiveness of AMiD. AMiD is primarily compared with methodologies, which inherently include the assistant distribution, such as GKD (Agarwal et al., 2024), DistiLLM (Ko et al., 2024), TAID (Shing et al., 2025); and the state-of-the-art ABKD (Wang et al., 2025). Please refer to Table 15 in Appendix C for further performance comparisons with additional baselines. We use α_{AB}-β_{AB}-divergence with $\alpha_{AB} = 0.2, \beta_{AB} = 0.7$ and adopt adaptive off-policy training (Ko et al., 2024) as default. We explore α over the range $\{-5, -3, -1, -0.5, 0, 0.5, 1.0\}$ and employ $\lambda = 0.1$ as default by following prior work (Ko et al., 2024). Additional details on datasets, models, and training details are provided in Appendix B.

4.1 Performance Comparison

Instruction-following Experiments. Table 1 reports results on the GPT-2 family (Radford et al., 2019) across different model sizes. AMiD consistently achieves the best performance in most evaluation settings, surpassing prior methods such as GKD, TAID, and DistiLLM, which also exploit assistant distributions. Notably, AMiD delivers substantial gains on SuperNI and UnNI, benchmarks requiring generalization to diverse and unseen instructions (Wang et al., 2022). These improvements suggest that AMiD promotes superior mode coverage and distributional alignment, thereby enhancing out-of-distribution generalization. Even when the capacity gap narrows for larger models (e.g., GPT-2 Large), AMiD continues to yield significant improvements, demonstrating that the α-mixture assistant distribution benefits not only small students but also stronger ones, thereby validating its scalability and robustness. Figure 4 further shows that AMiD consistently envelopes the baseline's

Table 4: ROUGE-L scores (↑) with various SGOs. We utilize GPT-2 XL (1.5B) → GPT-2 (0.1B). We use $\lambda = 0.1$ for these experiments. q_θ indicates no-assistant baseline ABKD for these experiments. We use D_{AB} for AMiD. $\alpha = -1$ and $\alpha = 1$ correspond to the assistant distribution employed in DistiLLM and TAID, respectively.

Dataset $\mathcal{D}$	Assistant $r_\theta^{(\alpha,\lambda)}$	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
Fixed (Hinton et al., 2015)	q_θ	27.06	24.81 ±0.28	11.25 ±0.26	15.05 ±0.21	21.78 ±0.15	23.73 ±0.09	19.32
	AMiD ($\alpha = -1$)	27.35	**25.34** ±0.28	11.54 ±0.26	15.34 ±0.30	21.77 ±0.19	24.45 ±0.06	19.69
	AMiD ($\alpha = 1$)	27.09	24.36 ±0.34	12.40 ±0.41	14.37 ±0.31	24.36 ±0.28	26.28 ±0.05	20.35
	AMiD ($\alpha \neq \pm 1$)	**27.84**	25.32 ±0.15	**13.44** ±0.41	**15.44** ±0.11	**27.03** ±0.12	**28.19** ±0.06	**21.88**
On-policy (Lin et al., 2020)	q_θ	28.25	25.70 ±0.20	13.03 ±0.17	16.86 ±0.34	24.67 ±0.16	27.47 ±0.15	21.55
	AMiD ($\alpha = -1$)	28.60	25.43 ±0.34	12.96 ±0.58	16.59 ±0.39	27.35 ±0.16	29.93 ±0.07	22.45
	AMiD ($\alpha = 1$)	28.60	25.12 ±0.55	13.28 ±0.38	17.08 ±0.52	24.35 ±0.23	26.94 ±0.19	21.35
	AMiD ($\alpha \neq \pm 1$)	**28.90**	**26.22** ±0.31	**14.31** ±0.22	**17.37** ±0.22	**28.59** ±0.27	**31.00** ±0.08	**23.50**
Mixed (Agarwal et al., 2024)	q_θ	29.08	25.67 ±0.16	12.38 ±0.29	**17.15** ±0.52	22.98 ±0.26	26.20 ±0.14	20.88
	AMiD ($\alpha = -1$)	28.79	25.65 ±0.25	11.98 ±0.28	16.94 ±0.13	23.82 ±0.17	26.25 ±0.06	20.93
	AMiD ($\alpha = 1$)	28.06	25.68 ±0.39	12.81 ±0.20	16.97 ±0.27	24.91 ±0.21	26.52 ±0.08	21.38
	AMiD ($\alpha \neq \pm 1$)	**29.24**	**26.46** ±0.16	**13.62** ±0.27	16.91 ±0.30	**28.13** ±0.06	**29.39** ±0.07	**22.90**
Adaptive off-policy (Ko et al., 2024)	q_θ	28.61	25.49 ±0.24	12.52 ±0.52	**17.36** ±0.55	26.07 ±0.14	27.36 ±0.10	21.76
	AMiD ($\alpha = -1$)	28.70	26.10 ±0.24	13.34 ±0.25	16.71 ±0.27	26.55 ±0.17	29.55 ±0.11	22.45
	AMiD ($\alpha = 1$)	27.80	25.78 ±0.44	13.74 ±0.19	16.42 ±0.22	26.04 ±0.22	27.79 ±0.09	21.95
	AMiD ($\alpha \neq \pm 1$)	**29.24**	**26.44** ±0.12	**13.74** ±0.49	16.76 ±0.24	**29.71** ±0.08	**30.35** ±0.09	**23.40**

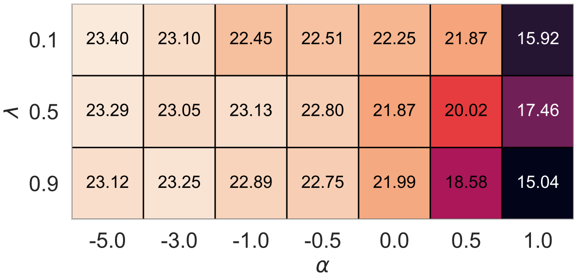


Figure 6: Relationship between α and λ under D_{AB}.

Table 5: Win Rate (WR, %) scores on three instruction-following benchmarks. We utilize Qwen2.5-14B-Instruct as a teacher and Qwen2.5-1.5B-Instruct as a student.

Model	AlpacaEval WR (↑)	Evol-Inst WR (↑)	UltraFeed WR (↑)	Avg. (↑)
Teacher	95.7	92.2	84.9	90.9
Student	64.3	47.2	40.4	50.6
DistiLLM-2 ($\alpha = -1$)	80.2	70.4	61.7	70.8
DistiLLM-2 ($\alpha = -5$)	**81.3**	**71.1**	**63.6**	**72.0**

validation ROUGE-L curve, indicating both efficient and stable optimization. Additional comparisons with other baselines and experiments on OpenLLaMA2 are provided in Appendix C.

Task-specific Experiments. To further investigate the effectiveness of AMiD, we consider various task-specific distillation, such as translation, summarization, and reasoning tasks. We adopt the implementation of SKD (Xu et al., 2024b) and employ the fixed dataset strategy. As shown in Table 2, using an assistant distribution achieves higher performance compared to the no-assistant baseline ABKD. Nevertheless, in the previous framework, where only $\alpha = \pm 1$ is available, it exhibits mixed performance depending on the task and network. Our proposed framework, AMiD, which extends the range of α, allows us to discover the high-performance assistant distribution. This generalization leads to consistent improvements over the baselines, achieving the best performance on all tasks.

4.2 Ablation Study and Additional Analysis

Balancing Mode-covering and Mode-seeking via α. As mentioned in Section 3.3, we have demonstrated that even when employing the same divergence, adjusting the α of the α-mixture assistant distribution allows us to control the mode-covering or mode-seeking property of the optimized student distribution. To further substantiate this analysis, we examine the trends of ROUGE-L, representing quality, and (Negative) Self-BLEU, representing diversity, by adjusting α with a fixed divergence. Figure 5 exhibits a clear quality-diversity trade-off for both KL divergence D_{KL} and α-β divergence D_{AB}. Specifically, as α increases, quality decreases and diversity increases, supporting the theoretical analysis that the mode-covering property is enhanced. Decreasing α shows the opposite effect, aligning with the mode of teacher distribution, which indicates mode-seeking. These results demonstrate that, even under a fixed divergence, α serves as an effective control knob to balance quality and diversity.

Relationship between α and λ. To further examine the relationship between α and λ, we conduct the experiments on $\lambda = 0.1, 0.5, 0.9$ with $D_{AB}(p||r_\theta^{(\alpha,\lambda)})$. Figure 6 presents the average performance of five instruction-following datasets among the various α and λ combinations. The analyses of experimental results are as follows: (1) Across all tested values of λ, using a smaller α consistently achieves higher performance. This observation aligns with our theoretical analysis, indicating

Table 6: Performances on instruction-following, code generation, and mathematical reasoning. We utilize the Qwen2.5-Instruct models, which are appropriate for each task, employing a 7B model as the teacher and a 1.5B model as the student. For evaluation, we employ LLM-as-a-Judge (Zheng et al., 2023) with GPT-4o for AlpacaEval and Evol-Instruct, GPT-4o-mini for UltraFeedback.

Model	Instruction-following				Code generation			Math reasoning
	AlpacaEval WR (↑)	Evol-Inst WR (↑)	UltraFeed WR (↑)	Avg. WR (↑)	HumanEval pass@1 (↑)	MBPP pass@1 (↑)	Avg. pass@1 (↑)	GSM8K (↑)
Teacher	93.7	89.6	80.8	88.0	90.9	83.1	87.0	89.3
Student	64.2	46.2	40.0	50.1	70.7	69.3	70.0	74.3
DistiLLM-2 ($\alpha = -1$)	79.5	69.0	62.6	70.4	72.0	74.6	73.3	76.9
DistiLLM-2 ($\alpha = -5$)	**80.7**	**71.0**	**63.3**	**71.7**	**73.2**	**73.5**	**73.4**	**77.4**

that a smaller α relatively induces a more mode-seeking behavior. (2) When λ is too large ($\lambda = 0.9$), performance slightly degrades and exhibits a larger standard deviation. We attribute this to the assistant distribution being overly close to the teacher distribution, which 1) limits effective knowledge transfer and 2) makes the optimization more sensitive to curvature variations. In contrast, $\lambda = 0.5$ achieves both high and stable performance, demonstrating that choosing a midpoint provides robustness against curvature changes. (3) Compared to λ, the performance is generally more robust to changes in α, as indicated by lower standard deviations. However, the α values close to 1 show noticeably higher standard deviations. We conjecture that this instability arises because the change of mixing coefficient (λ) between the teacher and student distributions makes hard mode-covering.

Compatibility with Divergences. As mentioned in Section 3.3, AMiD allows the use of the arbitrary divergence D since the α-mixture assistant distribution $r_\theta^{(\alpha,\lambda)}$ is a valid distribution. Therefore, we conduct performance comparisons across various combinations of divergences and α values to demonstrate the versatility of AMiD. In Table 3, we observed that AMiD generally achieves higher performance than the no-assistant baseline regardless of divergence in most settings. Notably, the highest-performing α was discovered in $\alpha \neq \pm 1$ regions beyond the limited scope of prior works, and its values are generally small. These results confirm the universality of AMiD to generic divergences, representing its wide adaptiveness and flexibility. Please refer to Appendix C.3 for the student-assistant cases $D(q_\theta, r_\theta^{(\alpha,\lambda)})$.

Universality to SGOs. AMiD is a generalized framework from the view of assistant distribution $r_\theta^{(\alpha,\lambda)}$ and divergence D, and therefore is not constrained by the dataset $\mathcal{D}$. In Table 4, we confirm the universality of AMiD across various student-generated output (SGO) strategies. AMiD ($\alpha \neq \pm 1$) outperforms the no-assistant baseline and previous mixtures ($\alpha = \pm 1$) by a significant margin across almost all metrics. These results indicate that AMiD is compatible with diverse SGO pipelines and remains effective regardless of how the datasets are collected.

Cooperation with Contrastive-based Distillation. The proposed α-mixture assistant distribution $r_\theta^{(\alpha,\lambda)}$ is a theoretically valid probability distribution for any α and λ. Therefore, the $r_\theta^{(\alpha,\lambda)}$ can be combined with the contrastive-based distillation methods beyond the divergence-based distillation methods. To validate this applicability, we replace the assistant distribution of DistiLLM-2 (Ko et al., 2025), which utilizes m-mixture ($\alpha = -1$), with our α-mixture assistant distribution. As shown in Table 6, incorporating our α-mixture assistant distribution exhibits further performance improvements or competitive performances over the base contrastive method, DistiLLM-2.

Scalability to Large Teacher. To demonstrate the extensibility, we conduct an experiment by distilling Qwen2.5-14B-Instruct into Qwen2.5-1.5B-Instruct under the instruction-following task. As shown in Table 5, integrating α-mixture assistant distribution exhibits performance improvements over the DistiLLM-2. These results indicate that AMiD remains effective in a large teacher model.

5 Conclusion

This work introduces a unified framework for KD in LLMs by proposing the α-mixture assistant distribution and the corresponding distillation method, AMiD. Our approach systematically generalizes previous fragmented methods and enables flexible interpolation between teacher and student. Theoretical and empirical analyses congruently demonstrate that the design parameter α controls the mode-seeking vs. mode-covering behavior. AMiD consistently outperforms prior KD methods across diverse settings and establishes a new foundation for assistant-guided KD for LLMs.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Shun-ichi Amari. Integration of stochastic models by minimizing α-divergence. *Neural computation*, 19(10):2780–2796, 2007.

Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Peter S Bullen. *Handbook of means and their inequalities*, volume 560. Springer Science & Business Media, 2013.

Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 5062–5074, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.449. URL https://aclanthology.org/2021.findings-acl.449/.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Andrzej Cichocki, Sergio Cruces, and Shun-ichi Amari. Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, 13(1):134–170, 2011.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instructiontuned llm. 2023.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Shinto Eguchi and Osamu Komori. Minimum divergence methods in statistical machine learning. *No Title*, 2022.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Roger B Grosse, Chris J Maddison, and Russ R Salakhutdinov. Annealing between distributions by averaging moments. *Advances in Neural Information Processing Systems*, 26, 2013.

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

GH Hardy. *Inequalities*. Cambridge University Press, 1952.

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

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806. URL https://aclanthology.org/2023.acl-long.806/.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, pp. 24872–24895. PMLR, 2024.

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

Andreĭ Nikolaevich Kolmogorov and Guido Castelnuovo. *Sur la notion de la moyenne*. G. Bardi, tip. della R. Accad. dei Lincei, 1930.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*, 2020.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023.

Vaden Masrani, Rob Brekelmans, Thang Bui, Frank Nielsen, Aram Galstyan, Greg Ver Steeg, and Frank Wood. q-paths: Generalizing the geometric annealing path using power means. In *Uncertainty in Artificial Intelligence*, pp. 1938–1947. PMLR, 2021.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10):1100, 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. *arXiv preprint arXiv:2501.16937*, 2025.

Zengkui Sun, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Warmup-distill: Bridge the distribution mismatch between teacher and student before knowledge distillation. *arXiv preprint arXiv:2502.11766*, 2025.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Guanghui Wang, Zhiyong Yang, Zitai Wang, Shi Wang, Qianqian Xu, and Qingming Huang. Abkd: Pursuing a proper allocation of the probability mass in knowledge distillation via α-β-divergence. *arXiv preprint arXiv:2505.04560*, 2025.

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 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. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340/.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. *arXiv preprint arXiv:2404.02657*, 2024.

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

Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024b.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. Lifting the curse of capacity gap in distilling language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4535–4553, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.249. URL https://aclanthology.org/2023.acl-long.249/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Definition and Proof

A.1 Amari's α-divergence

Definition. Let $\alpha \in \mathbb{R}$ be a real parameter. The α-function $f_\alpha(u)$ is defined for $\alpha \neq \pm 1$ by

$$f_\alpha(u) = \frac{4}{1-\alpha^2}\left(1 - u^{\frac{1+\alpha}{2}}\right). \tag{12}$$

Using this α-function, the α-divergence between two probability distributions $p = \{p_i\}$ and $q = \{q_i\}$ is defined as

$$D_\alpha[p||q] = \frac{4}{1-\alpha^2}\left(1 - \sum_i p_i^{\frac{1+\alpha}{2}} q_i^{\frac{1-\alpha}{2}}\right), \qquad \alpha \neq \pm 1. \tag{13}$$

The dual divergence is obtained by replacing α with $-\alpha$:

$$D_\alpha[p||q] = D_{-\alpha}[q||p]. \tag{14}$$

We also define the limiting cases at $\alpha = \pm 1$ by continuity. When $\alpha = \pm 1$, the α-function becomes

$$f_\alpha(u) = \begin{cases} u \log u, & \alpha = 1, \\ -\log u, & \alpha = -1. \end{cases} \tag{15}$$

Connection to KL and reverse KL. Correspondingly, the α-divergence reduces to

$$D_\alpha[p||q] = \begin{cases} \sum_i q_i \log \dfrac{q_i}{p_i}, & \alpha = 1, \\ \sum_i p_i \log \dfrac{p_i}{q_i}, & \alpha = -1, \end{cases} \tag{16}$$

which correspond to the KL divergence ($\alpha = -1$) and the reverse KL divergence ($\alpha = 1$).

Relation to Rényi divergence. Since the Amari α-divergence includes KL and reverse KL as the limiting cases, it can be viewed as a one-parameter generalization of the KL divergence. Moreover, it is closely related to the Rényi divergence. If we define a reparameterized order

$$\beta = \frac{1+\alpha}{2},$$

then the quantity appearing in the Amari divergence,

$$\sum_i p_i^{\frac{1+\alpha}{2}} q_i^{\frac{1-\alpha}{2}},$$

is exactly the same expression inside the discrete Rényi divergence,

$$R_\beta[p||q] = \frac{1}{\beta - 1} \log\left(\sum_i p_i^\beta q_i^{1-\beta}\right). \tag{17}$$

Thus, the Amari and Rényi divergences represent the same underlying one-parameter family through the invertible reparameterization $\beta = (1+\alpha)/2$

A.2 Proof of Proposition 3.1

Proposition 3.1. *The assistant distribution of TAID (Shing et al., 2025) is e-mixture of p and q_θ:*[3]

$$r_\theta := \mathrm{softmax}((1-\lambda) \cdot \mathrm{logit}(q_\theta) + \lambda \cdot \mathrm{logit}(p)) \propto p^\lambda q_\theta^{1-\lambda} \tag{6}$$

[3] We omit the time index t and detached notation for the sake of uniformity.

Proof. Note that $p = \mathrm{softmax}(\mathrm{logit}(p)) = \frac{1}{Z_p}\exp(\mathrm{logit}(p))$ where $\mathrm{logit}(p)$ is logit of p and Z_p is normalization constant of p. Therefore, $\mathrm{logit}(p) = \log(p \cdot Z_p)$.

$$\begin{aligned}
r &= \mathrm{softmax}(\lambda \mathrm{logit}(p) + (1-\lambda)\mathrm{logit}(q)) && (18)\\
&= \frac{1}{Z_r}\exp(\lambda \mathrm{logit}(p) + (1-\lambda)\mathrm{logit}(q)) && (19)\\
&= \frac{1}{Z_r}\exp(\lambda \log(pZ_p) + (1-\lambda)\log(qZ_q)) && (20)\\
&= \frac{1}{Z_r}\exp\left(\log\left(p^\lambda q^{1-\lambda}\right) + \log\left(Z_p^\lambda Z_q^{1-\lambda}\right)\right) && (21)\\
&= \frac{Z_p^\lambda Z_q^{1-\lambda}}{Z_r} p^\lambda q^{1-\lambda} && (22)\\
&= \frac{1}{Z'} p^\lambda q^{1-\lambda}, \quad \text{where } Z' := \frac{Z_r}{Z_p^\lambda Z_q^{1-\lambda}}. && (23)
\end{aligned}$$

Now, it is sufficient to show that $Z' = \sum p^\lambda q^{1-\lambda}$:

$$\begin{aligned}
Z' &= \frac{1}{Z_p^\lambda Z_q^{1-\lambda}} \sum \exp(\lambda \mathrm{logit}(p) + (1-\lambda)\mathrm{logit}(q)) && (24)\\
&= \sum \left(\frac{1}{Z_p}\exp(\mathrm{logit}(p))\right)^\lambda \left(\frac{1}{Z_q}\exp(\mathrm{logit}(q))\right)^{1-\lambda} && (25)\\
&= \sum p^\lambda q^{1-\lambda}. && (26)
\end{aligned}$$

Therefore, $r_\theta \propto p^\lambda q^{1-\lambda}$ and it is a valid distribution with normalization. □

A.3 Proof of Proposition 3.3

Proposition 3.3. *(Continuity) Assume that p and q_θ are not both zero. Then, $r_\theta^{(\alpha,\lambda)}$ is continuous function w.r.t α under the fixed $\lambda \in [0,1]$.*

Proof. We begin with the proof for the continuity of the unnormalized assistant distribution $\tilde{r}_\theta^{(\alpha,\lambda)}$. The $\alpha \neq 1$ case is trivial since it is a composition of continuous functions. For the $\alpha = 1$ case, it is a well-known fact that the power mean is a continuous function (Bullen, 2013). Especially, we can show that as follows:

$$\begin{aligned}
\lim_{\alpha\to 1} \log \tilde{r}_\theta^{(\alpha,\lambda)} &= \lim_{\alpha\to 1} \frac{2}{1-\alpha} \log\left(\lambda\, p^{\frac{1-\alpha}{2}} + (1-\lambda)\, q^{\frac{1-\alpha}{2}}\right) && (27)\\
&= \lim_{\alpha\to 1} \frac{\lambda\, p^{\frac{1-\alpha}{2}} \log p + (1-\lambda)\, q^{\frac{1-\alpha}{2}} \log q}{\lambda\, p^{\frac{1-\alpha}{2}} + (1-\lambda)\, q^{\frac{1-\alpha}{2}}} && (28)\\
&= \lambda \log p + (1-\lambda)\log q && (29)\\
&= \log\left(p^\lambda q^{1-\lambda}\right). && (30)
\end{aligned}$$

By the continuity of the exponential function, we can get $\lim_{\alpha\to 1} \tilde{r}_\theta^{(\alpha,\lambda)} = p^\lambda q^{1-\lambda}$. We use L'Hôpital's rule in the second equality. Note that $Z = \sum_i \tilde{r}_\theta^{(\alpha,\lambda)}(i)$ is continuous function w.r.t α since it is a finite sum of continuous functions w.r.t α. Also, since p and q cannot be both zero, $\tilde{r}_\theta^{(\alpha,\lambda)}$ is not zero, so $Z > 0$. Therefore, the $r_\theta^{(\alpha,\lambda)} = \frac{1}{Z}\tilde{r}_\theta^{(\alpha,\lambda)}$ is contiuous function w.r.t. α. □

A.4 Proof of Theorem 3.4

Theorem 3.4. *(Optimality) Let D be any proper divergence and $\alpha \in \mathbb{R}$.*

- *If $\lambda \in [0,1)$ and $\exists\theta$ s.t. $D(p, r_\theta^{(\alpha,\lambda)}) = 0$, then $D(p, r_\theta^{(\alpha,\lambda)}) = 0$ if and only if $p = q_\theta$.*
- *If $\lambda \in (0,1]$ and $\exists\theta$ s.t. $D(q_\theta, r_\theta^{(\alpha,\lambda)}) = 0$, then $D(q_\theta, r_\theta^{(\alpha,\lambda)}) = 0$ if and only if $p = q_\theta$.*

Proof. We first prove that $D(p, r_\theta^{(\alpha,\lambda)})$ implies $p = q_\theta$. By the definition of divergence, we have that $D(p, r_\theta^{(\alpha,\lambda)}) = 0$ if and only if $p = r_\theta^{(\alpha,\lambda)}$.

Case $\alpha = 1$. In this case, $r_\theta^{(\alpha,\lambda)} = \frac{1}{Z} p^\lambda q_\theta^{(1-\lambda)}$.

$$p = \frac{1}{Z} p^\lambda \, q_\theta^{1-\lambda} \Leftrightarrow Z \, p^{1-\lambda} = q_\theta^{1-\lambda} \Leftrightarrow Z^{\frac{1}{1-\lambda}} \, p = q_\theta \tag{31}$$

By integrating both sides, $Z^{\frac{1}{1-\lambda}} = 1$ which implies $Z = 1$. Therefore, $p = q_\theta$

Case $\alpha \neq 1$. In this case, $r_\theta^{(\alpha,\lambda)} = \frac{1}{Z}\left\{\lambda \, p^{\frac{1-\alpha}{2}} + (1-\lambda)\, q_\theta^{\frac{1-\alpha}{2}}\right\}^{\frac{2}{1-\alpha}}$

$$p = \frac{1}{Z}\left\{\lambda \, p^{\frac{1-\alpha}{2}} + (1-\lambda)\, q_\theta^{\frac{1-\alpha}{2}}\right\}^{\frac{2}{1-\alpha}} \Leftrightarrow Z^{\frac{1-\alpha}{2}} \, p^{\frac{1-\alpha}{2}} = \lambda \, p^{\frac{1-\alpha}{2}} + (1-\lambda)\, q_\theta^{\frac{1-\alpha}{2}} \tag{32}$$

$$\Leftrightarrow \left(Z^{\frac{1-\alpha}{2}} - \lambda\right) p^{\frac{1-\alpha}{2}} = (1-\lambda)\, q_\theta^{\frac{1-\alpha}{2}} \tag{33}$$

$$\Leftrightarrow C \, p^{\frac{1-\alpha}{2}} = q_\theta^{\frac{1-\alpha}{2}}, \quad \text{where} \quad C := \frac{Z^{\frac{1-\alpha}{2}} - \lambda}{1-\lambda} \tag{34}$$

$$\Leftrightarrow C^{\frac{2}{1-\alpha}} \, p = q_\theta \tag{35}$$

By integrating both sides, $C^{\frac{2}{1-\alpha}} = 1$ which implies $C = 1$. Therefore, $p = q_\theta$.

$D(q_\theta, r_\theta^{(\alpha,\lambda)})$ is similar. □

A.5 Proof of Proposition 3.5

Proposition 3.5. *(Gradient analysis) The gradient of f-divergence $D_f(p||r_\theta^{(\alpha,\lambda)})$ be expressed as:*

$$\nabla_\theta D_f\left(p||r_\theta^{(\alpha,\lambda)}\right) = \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[w \cdot \left\{\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right) - \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right)\right]\right\} \cdot \nabla_\theta \log q_\theta\right] \tag{11}$$

where $w := \frac{(1-\lambda) q_\theta^{\frac{1-\alpha}{2}}}{\lambda p^{\frac{1-\alpha}{2}} + (1-\lambda) q_\theta^{\frac{1-\alpha}{2}}}$ and $\psi_f(v) := f(v) - v f'(v)$.

Proof. From the basic calculus, we can derive the following equation for fixed p:

$$\frac{\partial}{\partial r}\left[r \, f\left(\frac{p}{r}\right)\right] = f\left(\frac{p}{r}\right) - \frac{p}{r} f'\left(\frac{p}{r}\right) = \psi_f\left(\frac{p}{r}\right). \tag{36}$$

Hence,

$$\nabla_\theta D_f(p||r_\theta^{(\alpha,\lambda)}) = \sum_{y_t \in \mathcal{V}} \psi_f\left(\frac{p(y_l \mid x, y_{<l})}{r_\theta^{(\alpha,\lambda)}(y_l \mid x, y_{<l})}\right) \cdot \nabla_\theta r_\theta^{(\alpha,\lambda)}(y_l \mid x, y_{<l}) \tag{37}$$

$$= \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right) \cdot \nabla_\theta \log r_\theta^{(\alpha,\lambda)}\right] \tag{38}$$

Before deriving the gradient of the log probability of $r_\theta^{(\alpha,\lambda)}$, let us first derive $\nabla_\theta \tilde{r}_\theta^{(\alpha,\lambda)}$ as follows:

$$\nabla_\theta \tilde{r}_\theta^{(\alpha,\lambda)} = \nabla_\theta \Big\{ h_\alpha^{-1}\big(\lambda\, h_\alpha(p) + (1-\lambda)\, h_\alpha(q_\theta)\big) \Big\} \tag{39}$$

$$= \frac{1}{h'_\alpha\big(\tilde{r}_\theta^{(\alpha,\lambda)}\big)}\,(1-\lambda)\,\nabla_\theta h_\alpha(q_\theta) \tag{40}$$

$$= \frac{1}{h'_\alpha\big(\tilde{r}_\theta^{(\alpha,\lambda)}\big)}\,(1-\lambda)\,h'_\alpha(q_\theta)\,\nabla_\theta q_\theta \tag{41}$$

$$= \frac{1}{h'_\alpha\big(\tilde{r}_\theta^{(\alpha,\lambda)}\big)}\,(1-\lambda)\,h'_\alpha(q_\theta)\,q_\theta\,\nabla_\theta \log q_\theta \tag{42}$$

$$= (1-\lambda)\,\frac{h'_\alpha(q_\theta)\,q_\theta}{h'_\alpha\big(\tilde{r}_\theta^{(\alpha,\lambda)}\big)\,\tilde{r}_\theta^{(\alpha,\lambda)}}\,\tilde{r}_\theta^{(\alpha,\lambda)}\,\nabla_\theta \log q_\theta \tag{43}$$

$$= (1-\lambda)\left(\frac{q_\theta}{\tilde{r}_\theta^{(\alpha,\lambda)}}\right)^{\frac{1-\alpha}{2}}\tilde{r}_\theta^{(\alpha,\lambda)}\,\nabla_\theta \log q_\theta \tag{44}$$

$$= \frac{(1-\lambda)\,q_\theta^{\frac{1-\alpha}{2}}}{\lambda\,p^{\frac{1-\alpha}{2}} + (1-\lambda)\,q_\theta^{\frac{1-\alpha}{2}}}\,\tilde{r}_\theta^{(\alpha,\lambda)}\,\nabla_\theta \log q_\theta \tag{45}$$

$$= w \cdot \tilde{r}_\theta^{(\alpha,\lambda)} \cdot \nabla_\theta \log q_\theta\,. \tag{46}$$

Therefore,

$$\nabla_\theta \log r_\theta^{(\alpha,\lambda)} = \frac{\nabla_\theta \tilde{r}_\theta^{(\alpha,\lambda)}}{\tilde{r}_\theta^{(\alpha,\lambda)}} - \frac{1}{Z_r}\sum_k \nabla_\theta \tilde{r}_\theta^{(\alpha,\lambda)}(k) \tag{47}$$

$$= w \cdot \nabla_\theta \log q_\theta - \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\big[w \cdot \nabla_\theta \log q_\theta\big] \tag{48}$$

Lastly, placing Eq. (48) into Eq. (38) and rearranging will yield the final result.

$$\nabla_\theta D_f(p\|r_\theta^{(\alpha,\lambda)}) = \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right)\cdot \nabla_\theta \log r_\theta^{(\alpha,\lambda)}\right] \tag{49}$$

$$= \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right)\cdot \Big\{w \cdot \nabla_\theta \log q_\theta - \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\big[w\cdot\nabla_\theta \log q_\theta\big]\Big\}\right] \tag{50}$$

$$= \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[w \cdot \left\{\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right) - \mathbb{E}_{r_\theta^{(\alpha,\lambda)}}\left[\psi_f\left(\frac{p}{r_\theta^{(\alpha,\lambda)}}\right)\right]\right\}\cdot \nabla_\theta \log q_\theta\right] \tag{51}$$

□

B Experimental Details

B.1 Toy experiment

We employ a two-modal Gaussian mixture for the teacher $p = 0.7\mathcal{N}(-3,2) + 0.3\mathcal{N}(3,0.8)$ and a unimodal Gaussian for the student $q_\theta = \mathcal{N}(\mu,\sigma^2)$ with $\mu_0 = 0, \sigma_0^2 = 1$. We optimize $\theta = \{\mu,\sigma^2\}$ by minizing $D_{\mathrm{KL}}(p\|r_\theta^{(\alpha,\lambda)})$ with Adam optimizer (Kingma & Ba, 2014) with 5000 steps and 5e-2 learning rate.

B.2 Datasets

- **databricks-dolly-15k** (Conover et al., 2023): An open-source dataset of instruction–response pairs created by thousands of Databricks employees. It covers diverse behavioral categories defined in Ouyang et al. (2022), including brainstorming, classification, closed QA, generation, information extraction, open QA, and summarization.
- **Self-instruct** (Wang et al., 2023): A framework for improving instruction-following ability by iteratively using model outputs to generate new instructional data. The dataset contains 52K instructions and 82K input–output pairs for tuning, 252 expert-written tasks for practical evaluation, and 50K additional examples from public datasets for benchmarking.
- **Vicuna** (Chiang et al., 2023): A benchmark consisting of 80 challenging open-ended questions originally used to assess Vicuna. It provides a compact but difficult testbed for evaluating instruction-following performance.
- **Super-Natural Instructions** (Wang et al., 2022): A large-scale benchmark comprising 1,616 expert-written NLP tasks spanning 76 task categories. Its test set includes 9K examples drawn from 119 tasks, covering a wide spectrum of instruction types.
- **Unnatural Instructions** (Honovich et al., 2023): An AI-generated dataset containing 240K instructions created with minimal human intervention. The collection demonstrates that synthetic data can serve as an effective substitute for human-curated data. Its core subset includes 60K examples.
- **AlpacaEval** (Dubois et al., 2023): AlpacaEval is derived from the AlpacaFarm evaluation suite but includes simplified formatting. In particular, Dubois et al. (2023) combined the original instruction and input fields into one unified instruction, a change that influences about a quarter of the samples sourced from Self-Instruct (Wang et al., 2023). The dataset ultimately comprises 805 difficult instruction-following queries.
- **Evol-Instruct Evaluation** (Xu et al., 2024a): This evaluation set includes 218 prompts produced through the Evol-Instruct generation pipeline. The questions span a wide range of topics and serve as a compact benchmark for instruction-following ability
- **UltraFeedback** (Cui et al., 2023): UltraFeedback is a large and detailed preference dataset tailored for building high-quality reward and critic models. It contains roughly 64k prompts collected from UltraChat, ShareGPT, and Evol-Instruct. Each prompt was used to elicit four responses from different LLMs, and GPT-4 subsequently annotated these outputs based on dimensions such as following instructions, factual correctness, honesty, and helpfulness.
- **MetaMathQA** (Yu et al., 2023): MetaMathQA was created to strengthen mathematical reasoning in LLMs. The dataset is generated through a bootstrapping method in which each math question is re-expressed using multiple reasoning viewpoints, including forward reasoning, backward reasoning, and paraphrased formulations.
- **GSM8K** (Cobbe et al., 2021): GSM8K consists of 8.5K meticulously written grade-school math word problems. Designed to require multi-step inference, it is widely used as a standard benchmark for evaluating basic mathematical reasoning skills.
- **WizardCoder** (Luo et al., 2023): WizardCoder is an instruction-tuned code dataset built via the Evol-Instruct method. Starting with the 20K-sample Code Alpaca corpus, the creators iteratively evolved prompts by increasing complexity, adding constraints, inserting misleading code, and introducing time/space complexity requirements. The final dataset contains about 78K evolved examples, which were used to fine-tune StarCoder and substantially improve its coding performance.
- **HumanEval** (Chen, 2021): HumanEval provides 164 hand-crafted programming tasks with function signatures, natural-language descriptions, and unit tests. It is one of the primary benchmarks for assessing code generation and was explicitly designed to avoid overlap with existing training data.
- **MBPP** (Austin et al., 2021): MBPP includes roughly 1,000 Python programming exercises aimed at novice programmers. Each problem comes with a textual description, a reference code solution, and three automatic test cases. Portions of the dataset were manually validated to ensure consistency and correctness.

B.3 Implementation Settings

Training. For instruction-following distillation, we use databricks-dolly-15K (Conover et al., 2023) for the distillation loss and OpenWebText (Gokaslan & Cohen, 2019) for the pretraining loss. Teacher models include GPT-2 XL (1.5B) with SFT, and students are GPT-2 (0.1B), GPT-2 Medium (0.3B), and GPT-2 Large (0.8B). To test scalability, OpenLLaMA2-7B (Geng & Liu, 2023) is distilled into OpenLLaMA2-3B using LoRA.

For task-specific evaluation, we use Flores-200 (Costa-Jussà et al., 2022) for translation, DialogSum (Chen et al., 2021) for summarization, and GSM8K (Cobbe et al., 2021) for mathematical reasoning. Teacher models are Gemma-7B-It (Team et al., 2024) and Qwen2-7B-Instruct (Team, 2024), while Gemma-2B-It and Qwen2-0.5B-Instruct serve as students. Teachers are fine-tuned on the full dataset, while students are trained with about 1,000 samples.

We use α_{AB}-β_{AB}-divergence with $\alpha_{AB} = 0.2, \beta_{AB} = 0.7$ and adopt adaptive off-policy training (Ko et al., 2024) as default. We explore α over the range $\{-5, -3, -1, -0.5, 0, 0.5, 1.0\}$ and employ $\lambda = 0.1$ as default by following prior work (Ko et al., 2024). We utilize the AdamW optimizer and cosine learning rate scheduling by following the previous work (Ko et al., 2024; Wang et al., 2025). We search the learning rate over the range $\{0.0005, 0.0001, 0.00005\}$ except for the cooperation with contrastive-based distillation experiments, which use the default setting of DistiLLM-2 (Ko et al., 2025).

Theoretical insights based α tuning guidelines. Herein, we provide a principled tuning guidelines grounded in the theoretical properties of α. First, in Section 3.2, we show that the support of the assistant distribution varies with α: $supp(r_\theta^{(\alpha,\lambda)}) = supp(p) \cup supp(q_\theta)$ when $\alpha < 1$, $supp(r_\theta^{(\alpha,\lambda)}) = supp(p) \cap supp(q_\theta)$ when $\alpha \geq 1$. In KD for LLMs, the teacher and student often exhibit a capacity gap and produce high-dimensional outputs with many near-zero probabilities. As a result, they do not share a sufficiently large common support in general. For this reason, we recommend using $\alpha < 1$ in most practical settings, as this choice improves training stability and enables more reliable knowledge transfer.

Furthermore, our gradient analysis and (toy) experiments demonstrate that α directly controls the trade-off between mode-covering and mode-seeking behavior of the optimized student. Under the $\alpha < 1$, increasing α encourages relatively stronger mode covering, thereby improving output diversity. Conversely, smaller α emphasizes mode seeking, which enhances fidelity to the teacher. Therefore, for enhancing the teacher–student alignment and performance, we suggest using small α values. However, since too small α can induce high curvature in the geometry of the interpolation path, which may reduce optimization efficiency, so such choices should be used with caution.

Based on these theoretical insights, we explore α over the range $\{-5, -3, -1, -0.5, 0, 0.5, 1.0\}$. Table 7 provides the detailed configuration for each task.

Table 7: Configuration of hyperparameters for AMiD

Task	**Dataset**	**Teacher**	**Student**	**Divergence**	α
Instruction following	databricks-dolly-15k	GPT-2 XLarge	GPT-2 Base	$D_{KL}(p\|\|r)$ $D_{RKL}(p\|\|r)$ $D_{AB}(p\|\|r)$ $D_{KL}(q\|\|r)$	-5.0 -3.0 -5.0 0.5
			GPT-2 Medium GPT-2 Large	$D_{AB}(p\|\|r)$ $D_{AB}(p\|\|r)$	-5.0 -3.0
		OpenLLaMA2-7B	OpenLLaMA2-3B	$D_{AB}(p\|\|r)$	-3.0
	UltraChat200k	Qwen2.5-7B-Instruct Qwen2.5-14B-Instruct	Qwen2.5-1.5B-Instruct Qwen2.5-1.5B-Instruct	$D_{DistilLLM-2}$ $D_{DistilLLM-2}$	-5.0 -5.0
Translation	Flores-200	Gemma-7B-It Qwen2-7B-Instruct	Gemma-2B-It Qwen2-0.5B-Instruct	$D_{AB}(p\|\|r)$ $D_{AB}(p\|\|r)$	-0.5 0.5
Summarization	DialogSum	Gemma-7B-It Qwen2-7B-Instruct	Gemma-2B-It Qwen2-0.5B-Instruct	$D_{AB}(p\|\|r)$ $D_{AB}(p\|\|r)$	0.5 -3.0
Mathematical reasoning	GSM8k	Gemma-7B-It Qwen2-7B-Instruct	Gemma-2B-It Qwen2-0.5B-Instruct	$D_{AB}(p\|\|r)$ $D_{AB}(p\|\|r)$	0.5 -3.0
	MetaMathQA	Qwen2.5-Math-7B-Instruct	Qwen2.5-Math-1.5B-Instruct	$D_{DistilLLM-2}$	-5.0
Code generation	WizardCoder	Qwen2.5-Coder-7B-Instruct	Qwen2.5-Coder-1.5B-Instruct	$D_{DistilLLM-2}$	-5.0

Evaluation. For evaluating generation quality, we adopt ROUGE-L (Lin, 2004) and Self-BLEU (Zhu et al., 2018). ROUGE-L measures the similarity between the generated output and the reference text by computing the Longest Common Subsequence (LCS). Specifically, recall and precision are defined as

$$R_{\text{LCS}} = \frac{LCS(x, y)}{L_x}, P_{\text{LCS}} = \frac{LCS(x, y)}{L_y}, \tag{52}$$

where $LCS(x, y)$ is the length of the longest common subsequence between the reference x and the generated text y, and L_x, L_y denote their respective lengths. The final ROUGE-L score is given by the harmonic mean:

$$ROUGE\text{-}L = \frac{2 \cdot R_{\text{LCS}} \cdot P_{\text{LCS}}}{R_{\text{LCS}} + P_{\text{LCS}}}. \tag{53}$$

A higher ROUGE-L score indicates that the generated text more closely matches the reference in terms of sequence overlap.

Self-BLEU evaluates the diversity of generated outputs by leveraging the BLEU metric (Papineni et al., 2002). BLEU computes the geometric mean of modified n-gram precisions with a brevity penalty (BP):

$$\text{BP} = \begin{cases} 1 & \text{if } c > r, \\ e^{(1-r/c)} & \text{if } c \leq r, \end{cases} \tag{54}$$

$$\text{BLEU}(c, R) = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n(c, R)\right), \tag{55}$$

where c is the candidate length, r is the effective reference length, $p_n(c, R)$ denotes the modified n-gram precision, and w_n are positive weights summing to one. Building on this definition, Self-BLEU is calculated by treating each generated sample s_i as the hypothesis and the remaining set $S \backslash \{s_i\}$ as references:

$$\text{Self-BLEU}(S) = \frac{1}{M} \sum_{i=1}^{M} \text{BLEU}(s_i, S \backslash \{s_i\}). \tag{56}$$

A higher Self-BLEU score (close to 1) indicates that the outputs are highly similar to each other, reflecting low diversity and more deterministic behavior, while a lower score (close to 0) suggests greater diversity across generations.

C Additional Experimental Results and Discussions

C.1 More comparison with baselines

Table 15 presents the complete results on the GPT-2 family, where we extend the comparison to a broader set of baseline methods beyond those reported in the main paper. We observe that AMiD consistently outperforms all competing approaches across different student sizes (0.1B, 0.3B, 0.8B), further validating the robustness of our method. In particular, while methods such as SeqKD, ImitKD, MiniLLM, and AKL yield modest improvements over standard knowledge distillation (KD), they still fall short of strong assistant-based methods like GKD, TAID, and DistiLLM. Among these baselines, ABKD often emerges as the strongest competitor. Nevertheless, AMiD achieves clear performance gains over ABKD in nearly every evaluation setting.

C.2 Results on OpenLLaMA2

Table 8 reports results on the OpenLLaMA2 family, where a 7B teacher is distilled into a 3B student. Consistent with our findings on the GPT-2 series, AMiD achieves the best overall performance across most evaluation benchmarks. In particular, AMiD surpasses prior assistant-based approaches such as TAID and DistiLLM (both SKL and SRKL variants), as well as the strong baseline ABKD.

Table 8: ROUGE-L scores (↑) on OpenLLaMA2-7B → OpenLLaMA2-3B. **Bold** and Underline mean the best and second-best performance of each column, except the teacher, respectively. All results are based on our own re-implementation. We conduct the evaluation with five random seeds.

Model	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
Teacher	–	27.60 $\pm$0.34	18.17 $\pm$0.80	17.85 $\pm$0.48	31.05 $\pm$0.31	32.40 $\pm$0.28	25.41
OpenLLaMA2-7B → OpenLLaMA2-3B							
TAID	30.85	26.53 $\pm$0.23	17.73 $\pm$0.69	18.14 $\pm$0.39	31.93 $\pm$0.23	31.55 $\pm$0.12	25.18
DistiLLM (SKL)	33.07	28.63 $\pm$0.28	20.20 $\pm$0.66	19.15 $\pm$0.32	35.31 $\pm$0.19	34.74 $\pm$0.10	27.61
DistiLLM (SRKL)	33.18	28.83 $\pm$0.41	20.76 $\pm$0.37	19.37 $\pm$0.15	36.82 $\pm$0.14	35.76 $\pm$0.13	28.31
ABKD	33.91	29.43 $\pm$0.42	20.46 $\pm$0.28	20.42 $\pm$0.12	**39.51** $\pm$0.25	**38.07** $\pm$0.08	29.58
AMiD (Ours)	**34.39**	**29.69** $\pm$0.47	**20.99** $\pm$0.37	**21.03** $\pm$0.40	39.06 $\pm$0.21	37.31 $\pm$0.11	**29.62**

C.3 Compatibility with Divergences $D_{\mathrm{KL}}(q_\theta \| r_\theta^{(\alpha,\lambda)})$

Table 9 provides the complementary results when employing the divergence $D_{\mathrm{KL}}(q_\theta \| r_\theta^{(\alpha,\lambda)})$, contrasting the student distribution against the α-mixture assistant. Similar to the findings in the main text (Table 3), AMiD consistently outperforms the no-assistant baseline across most evaluation benchmarks, confirming that the proposed method is broadly compatible with different divergence directions.

Table 9: ROUGE-L scores (↑) with $D_{\mathrm{KL}}(q_\theta \| r_\theta^{(\alpha,\lambda)})$ and various α. We utilize GPT-2 XL (1.5B) → GPT-2 (0.1B). We use a fixed $\lambda = 0.9$ for these experiments.

Divergence D	Assistant $r_\theta^{(\alpha,\lambda)}$	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
$D_{\mathrm{KL}}(q_\theta \| r_\theta^{(\alpha,\lambda)})$	q_θ	28.71	26.22 $\pm$0.35	12.57 $\pm$0.16	16.97 $\pm$0.34	24.75 $\pm$0.20	26.59 $\pm$0.14	21.42
	AMiD ($\alpha = -5.0$)	27.54	24.23 $\pm$0.23	12.59 $\pm$0.32	15.80 $\pm$0.43	24.50 $\pm$0.14	26.38 $\pm$0.07	20.70
	AMiD ($\alpha = -3.0$)	27.96	25.13 $\pm$0.29	12.80 $\pm$0.48	16.32 $\pm$0.45	25.54 $\pm$0.30	26.86 $\pm$0.14	21.33
	AMiD ($\alpha = -1.0$)	28.21	25.74 $\pm$0.20	12.13 $\pm$0.23	16.34 $\pm$0.15	25.40 $\pm$0.10	26.91 $\pm$0.12	21.30
	AMiD ($\alpha = -0.5$)	28.89	26.01 $\pm$0.32	12.84 $\pm$0.59	**17.04** $\pm$0.05	**27.43** $\pm$0.14	27.59 $\pm$0.03	22.18
	AMiD ($\alpha = 0.0$)	28.84	**26.70** $\pm$0.33	13.36 $\pm$0.31	15.95 $\pm$0.36	26.23 $\pm$0.17	27.70 $\pm$0.10	21.99
	AMiD ($\boldsymbol{\alpha = -0.5}$)	**29.02**	26.48 $\pm$0.17	**13.73** $\pm$0.44	16.78 $\pm$0.30	26.78 $\pm$0.34	**28.65** $\pm$0.10	**22.48**
	AMiD ($\alpha = 1.0$)	28.40	26.01 $\pm$0.34	12.03 $\pm$0.33	16.96 $\pm$0.31	24.84 $\pm$0.24	27.01 $\pm$0.11	21.37

C.4 Overlap-based Adaptive α scheduling

The theoretical insights based α tuning guidelines efficiently exclude low potential candidates. However, applying a single fixed global α value can still be sub-optimal in certain cases.

To address this concern, we introduce a curriculum-based adaptive α scheduling based on the degree of overlap between token-level teacher distribution $p(y_l|y_{<l}, x)$ and student distribution $q_\theta(y_l|y_{<l}, x)$. The intuition is that when the teacher and student distributions are highly overlapped, we encourage mode-covering to align further, whereas when the overlap is low, we enhance mode-seeking to find the mode first.

We define token-level overlap as $ovl_{i,l} := \sum_{y_l} \min(p(y_l|y_{<l}, x), q_\theta(y_l|y_{<l}, x))$. Obviously, the overlap value $ovl_{i,l}$ is bounded into $[0, 1]$. Also, $ovl_{i,l}$ approaches 0 when $p(y_l|y_{<l}, x)$ and $q_\theta(y_l|y_{<l}, x)$ significantly differ, and approaches 1 when they are well-aligned. Furthermore, $ovl_{i,l}$ can be expressed in terms of total variation distance $1 - TVD(p(y_l|y_{<l}, x), q_\theta(y_l|y_{<l}, x))$, which provides same interpretation.

Given the predefined α_{min} and α_{max}, we set the token-level $\alpha_{i,l}$ as the linearly increasing value along the line passing through $(0, \alpha_{min})$ and $(1, \alpha_{max})$ i.e. $\alpha_{i,l} \leftarrow (\alpha_{max} - \alpha_{min}) * ovl_{i,l} + \alpha_{min}$. Under this idea, when the teacher and student distributions differ substantially, the $ovl_{i,l}$ becomes small, leading to a smaller assigned α, which strengthens mode-seeking. Conversely, when the teacher and student distributions are similar, both $ovl_{i,l}$ and α have larger values, thereby reinforcing mode-covering. This mechanism systematically determines α by combining the degree of alignment between the teacher and student distribution, which continuously changes through training, with the theoretical characteristics of α.

Table 10: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets with fixed α versus adaptive α scheduling. **Bold** means the best performance of each column. We use D_{AB} and $\lambda = 0.1$ for AMiD.

Assistant	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
AMiD (Fixed α)	29.24	26.44	13.74	16.76	29.71	30.35	23.40
AMiD (Adaptive α)	29.31	26.50	14.02	16.59	29.87	30.60	**23.52**

C.5 Robustness to Optimizer

To investigate whether the effectiveness of AMiD depends on a particular optimization setup, we evaluate its performance under the Lion optimizer. Table 11 exhibits the robustness of AMiD w.r.t. the optimizer.

Table 11: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets under the Lion optimizer across different α-mixture configurations. **Bold** means the best performance of each column. We use D_{AB} and $\lambda = 0.1$ for AMiD.

	Assistant	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
Lion	No assistant	28.38	26.02 $\pm$0.28	12.19 $\pm$0.34	17.24 $\pm$0.37	26.29 $\pm$0.19	28.88 $\pm$0.10	22.12
	AMiD ($\alpha = -1$)	28.68	25.29 $\pm$0.16	12.21 $\pm$0.25	17.81 $\pm$0.36	24.82 $\pm$0.13	27.87 $\pm$0.08	21.60
	AMiD ($\alpha = +1$)	24.98	22.33 $\pm$0.27	9.50 $\pm$0.28	15.50 $\pm$0.52	15.71 $\pm$0.34	18.01 $\pm$0.08	16.21
	AMiD ($\alpha = \pm 1$)	27.85	26.14 $\pm$0.21	12.55 $\pm$0.12	17.25 $\pm$0.48	28.28 $\pm$0.26	29.69 $\pm$0.06	**22.78**

C.6 Robustness to Learning rate scheduling

To further assess the robustness of AMiD to optimization hyperparameters, we evaluate its performance under the Noam learning rate schedule, originally designed for transformer architectures. Table 12 also support the robustness of AMiD.

Table 12: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets under the Noam learning rate schedule across different α-mixture configurations. **Bold** means the best performance of each column. We use D_{AB} and $\lambda = 0.1$ for AMiD.

	Assistant	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
Noam	No assistant	28.42	25.83 $\pm$0.17	13.35 $\pm$0.54	16.43 $\pm$0.28	28.30 $\pm$0.24	29.90 $\pm$0.11	22.76
	AMiD ($\alpha = -1$)	28.91	26.02 $\pm$0.34	14.07 $\pm$0.25	17.09 $\pm$0.19	27.78 $\pm$0.11	29.49 $\pm$0.04	22.93
	AMiD ($\alpha = +1$)	28.33	25.59 $\pm$0.25	13.61 $\pm$0.49	16.25 $\pm$0.34	26.42 $\pm$0.20	28.26 $\pm$0.19	22.09
	AMiD ($\alpha = \pm 1$)	29.39	26.12 $\pm$0.35	13.07 $\pm$0.52	16.53 $\pm$0.46	29.06 $\pm$0.14	30.86 $\pm$0.09	**23.19**

C.7 Mitigate the conflict via Temperature Scaling

As discussed in Section 3.2, combining an assistant distribution with a narrow support and a divergence that requires the expectation w.r.t. the assistant distribution can lead to training instability and poor knowledge transfer.

Since the primary cause of this issue is the narrow support of the assistant distribution, we conjecture that applying distribution softening technique could alleviate the instability even for such problematic combinations. To verify this conjecture, we employ temperature $T > 1$, which is a widely used flattening technique in various area. The table below shows the performance when using various temperature values under $D_{RKL}(p||r_\theta^{(\alpha,\lambda)})$ with $\alpha = 1$. The results exhibit that introducing the temperature leads to stable training and can even yield strong performance with an appropriately chosen temperature value. However, large temperature causes an over-flattening effect, inducing large shift in the assistant distribution and consequently degrading performance. Overall, we demonstrate that temperature scaling can empirically mitigate the instability associated with problematic combinations under the appropriate temperature value.

Table 13: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets under $D_{\text{RKL}}(p||r_\theta^{(\alpha,\lambda)})$ with $\alpha = 1$ when applying temperature scaling to soften the assistant distribution.

Assistant	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
AMiD ($\alpha = 1.0, T = 1.0$)	0.16	4.27	2.81	9.12	1.64	1.84	3.94
AMiD ($\alpha = 1.0, T = 1.5$)	27.40	24.61	11.55	17.11	21.41	23.39	19.61
AMiD ($\alpha = 1.0, T = 2.0$)	28.64	26.83	12.74	17.62	23.56	27.01	21.55
AMiD ($\alpha = 1.0, T = 5.0$)	28.78	26.78	12.43	17.30	25.87	27.74	22.02
AMiD ($\alpha = 1.0, T = 10.0$)	27.33	24.73	12.11	16.95	23.65	26.80	20.85

Table 14: ROUGE-L scores (↑) across different combinations of λ and the α-mixture assistant distribution.

λ	Assistant	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
	No Assistant	28.61	25.49	12.52	17.36	26.07	27.36	21.76
0.1	AMiD ($\alpha = -5.0$)	29.24	26.44	13.74	16.76	29.71	30.35	**23.40**
	AMiD ($\alpha = -3.0$)	29.07	26.38	13.58	16.11	29.27	30.14	23.10
	AMiD ($\alpha = -1.0$)	28.70	26.10	13.34	16.71	26.55	29.55	22.45
	AMiD ($\alpha = -0.5$)	28.70	26.37	13.59	17.02	27.06	28.50	22.51
	AMiD ($\alpha = 0.0$)	28.86	25.77	13.57	16.14	27.26	28.52	22.25
	AMiD ($\alpha = 0.5$)	28.46	25.80	12.94	16.59	26.29	27.73	21.87
	AMiD ($\alpha = 1.0$)	24.93	22.36	9.72	16.29	15.09	16.15	15.92
0.5	AMiD ($\alpha = -5.0$)	29.38	26.41	13.81	16.44	29.19	30.58	**23.29**
	AMiD ($\alpha = -3.0$)	29.38	26.45	13.71	16.43	28.23	30.44	23.05
	AMiD ($\alpha = -1.0$)	29.11	26.31	14.09	16.70	28.68	29.89	23.13
	AMiD ($\alpha = -0.5$)	28.79	26.55	13.85	16.30	27.85	29.46	22.80
	AMiD ($\alpha = 0.0$)	28.74	25.68	13.01	16.51	26.32	27.83	21.87
	AMiD ($\alpha = 0.5$)	26.73	24.13	12.06	16.19	22.88	24.82	20.02
	AMiD ($\alpha = 1.0$)	22.88	21.31	10.41	14.87	19.35	21.34	17.46
0.9	AMiD ($\alpha = -5.0$)	29.26	26.64	13.49	16.40	28.65	30.40	23.12
	AMiD ($\alpha = -3.0$)	29.14	26.02	13.62	17.03	28.99	30.59	**23.25**
	AMiD ($\alpha = -1.0$)	29.32	26.19	13.24	15.83	29.23	29.97	22.89
	AMiD ($\alpha = -0.5$)	29.12	26.19	13.32	16.40	28.15	29.68	22.75
	AMiD ($\alpha = 0.0$)	28.28	25.23	13.01	15.51	27.52	28.66	21.99
	AMiD ($\alpha = 0.5$)	22.05	19.92	10.76	12.33	25.09	24.78	18.58
	AMiD ($\alpha = 1.0$)	21.52	19.17	9.16	13.60	15.53	17.73	15.04

D Discussion of Optimality

Theorem 3.4 guarantees the optimality of AMiD, yet experimentally demonstrated extremely poor performance for the reverse KL divergence $D_{\text{RKL}}(p\|r_\theta^{(\alpha,\lambda)})$ and $\alpha = 1$ in Table 3. We conjecture that it is caused by the conflict between RKL and the support intersection property, which leads to instability. RKL includes the expectation of the assistant distribution $\mathbb{E}_{r_\theta^{(\alpha,\lambda)}}[\cdot]$ by definition. However, when $\alpha = 1$, since $\text{supp}(r_\theta^{(\alpha,\lambda)})$ is $\text{supp}(p) \cap \text{supp}(q_\theta)$ (see Section 3.2), $\mathbb{E}_{r_\theta^{(\alpha,\lambda)}}[\cdot]$ is conducted on an unstable and narrow region, and this phenomenon intensifies further in the early stages of optimization. In addition, we experimentally find that the combination of $D_{\text{RKL}}(p\|r_\theta^{(\alpha,\lambda)})$ and $\alpha = 1$ produces highly unstable loss and gradient within a few early steps. In conclusion, while AMiD theoretically guarantees optimality, it might be necessary to employ appropriate divergence and alpha values, taking into account the imperfect optimization.

E The Use of Large Language Models (LLMs)

We employed the LLM to polish the paper writing. Specifically, it was used to request grammatical corrections once the author had drafted the text.

Table 15: ROUGE-L scores (↑) on five task-agnostic instruction-following datasets. **Bold** and Underline mean the best and second-best performance of each column, except the teacher, respectively. All results are based on our own re-implementation. We conduct the evaluation with five random seeds.

Model	Val. (↑)	Dolly Eval (↑)	Self Inst (↑)	Vicuna (↑)	Super NI (↑)	UnNI (↑)	Avg. (↑)
Teacher	–	27.14 ±0.15	14.55 ±0.82	16.12 ±0.31	27.21 ±0.25	31.41 ±0.06	23.29
GPT-2 XL (1.5B) → GPT-2 (0.1B)							
SFT	25.81	23.54 ±0.42	9.62 ±0.21	14.79 ±0.56	18.42 ±0.23	19.33 ±0.13	17.14
KD	25.25	23.44 ±0.33	10.12 ±0.28	14.93 ±0.29	16.88 ±0.24	18.87 ±0.16	16.85
SeqKD	26.07	24.20 ±0.31	11.12 ±0.09	15.82 ±0.37	19.29 ±0.09	22.74 ±0.05	18.63
ImitKD	23.91	22.02 ±0.29	10.34 ±0.53	15.32 ±0.26	17.34 ±0.26	19.68 ±0.15	16.94
GKD	27.06	24.58 ±0.13	11.78 ±0.44	14.60 ±0.37	22.84 ±0.12	25.04 ±0.09	19.77
MiniLLM	-	24.47 ±0.18	12.83 ±0.50	16.94 ±0.40	25.58 ±0.33	26.38 ±0.17	21.24
AKL	25.62	23.23 ±0.35	11.18 ±0.21	14.94 ±0.23	19.36 ±0.39	22.41 ±0.08	18.22
TAID	28.37	25.74 ±0.27	12.91 ±0.31	17.09 ±0.18	23.66 ±0.31	26.82 ±0.05	21.24
DistiLLM (SKL)	27.88	25.50 ±0.28	12.35 ±0.39	16.10 ±0.22	23.87 ±0.39	26.16 ±0.06	20.80
DistiLLM (SRKL)	28.21	25.74 ±0.20	12.13 ±0.23	16.34 ±0.15	25.40 ±0.10	26.91 ±0.12	21.30
ABKD	28.61	25.49 ±0.24	12.52 ±0.52	**17.36** ±0.55	26.07 ±0.14	27.36 ±0.10	21.76
AMiD (Ours)	**29.24**	**26.44** ±0.12	**13.74** ±0.49	16.76 ±0.24	**29.71** ±0.08	**30.35** ±0.09	**23.40**
GPT-2 XL (1.5B) → GPT-2 Medium (0.3B)							
SFT	27.96	25.70 ±0.35	12.60 ±0.37	16.51 ±0.19	24.21 ±0.13	27.51 ±0.17	21.31
KD	26.03	24.27 ±0.42	10.58 ±0.10	15.59 ±0.10	18.15 ±0.13	20.49 ±0.24	17.82
SeqKD	28.41	26.61 ±0.34	13.01 ±0.46	16.42 ±0.63	23.44 ±0.20	26.93 ±0.08	21.28
ImitKD	25.93	24.46 ±0.62	12.00 ±0.41	15.56 ±0.46	20.12 ±0.34	25.11 ±0.16	19.45
GKD	27.90	25.06 ±0.55	12.36 ±0.42	15.71 ±0.58	23.83 ±0.26	27.14 ±0.09	20.82
MiniLLM	-	25.80 ±0.57	14.87 ±0.35	17.62 ±0.33	26.78 ±0.26	30.70 ±0.11	23.15
AKL	27.81	25.57 ±0.10	12.06 ±0.56	15.98 ±0.17	22.22 ±0.20	26.17 ±0.13	20.40
TAID	29.45	27.01 ±0.27	14.53 ±0.47	17.58 ±0.20	25.14 ±0.15	29.79 ±0.14	22.81
DistiLLM (SKL)	29.65	26.87 ±0.13	14.11 ±0.29	16.85 ±0.54	25.59 ±0.22	28.84 ±0.03	22.45
DistiLLM (SRKL)	29.72	26.50 ±0.20	13.79 ±0.71	17.14 ±0.52	26.25 ±0.11	29.31 ±0.16	22.60
ABKD	29.64	26.93 ±0.17	13.69 ±0.32	17.45 ±0.27	28.15 ±0.18	30.94 ±0.06	23.43
AMiD (Ours)	**30.83**	**27.34** ±0.18	**15.26** ±0.46	**17.69** ±0.27	**29.04** ±0.20	**33.15** ±0.13	**24.50**
GPT-2 XL (1.5B) → GPT-2 Large (0.8B)							
SFT	28.48	26.17 ±0.41	13.78 ±0.21	16.64 ±0.48	23.76 ±0.30	26.64 ±0.12	21.40
KD	28.52	26.27 ±0.26	13.72 ±0.44	16.43 ±0.25	25.24 ±0.18	28.94 ±0.09	22.12
SeqKD	28.24	26.16 ±0.41	13.93 ±0.56	16.35 ±0.20	25.03 ±0.27	28.58 ±0.06	22.01
ImitKD	26.96	23.37 ±0.40	13.26 ±0.60	16.00 ±0.33	23.31 ±0.16	27.59 ±0.14	20.71
GKD	29.36	26.38 ±0.24	14.44 ±0.66	17.02 ±0.46	26.64 ±0.16	30.99 ±0.13	23.09
MiniLLM	-	26.30 ±0.35	16.50 ±0.52	18.14 ±0.49	29.45 ±0.17	34.40 ±0.17	24.96
AKL	27.69	25.45 ±0.40	13.83 ±0.82	15.85 ±0.35	25.41 ±0.25	28.91 ±0.05	21.89
TAID	29.83	26.85 ±0.32	15.07 ±0.31	17.02 ±0.48	26.71 ±0.23	31.09 ±0.17	23.35
DistiLLM (SKL)	29.69	26.12 ±0.27	15.69 ±0.75	16.91±0.43	27.23 ±0.18	30.73 ±0.12	23.34
DistiLLM (SRKL)	30.59	27.09 ±0.40	14.61 ±0.66	16.39 ±0.27	28.44 ±0.45	31.04 ±0.06	23.51
ABKD	30.49	27.67 ±0.34	15.46 ±0.81	**17.43** ±0.25	30.74 ±0.22	33.11 ±0.15	24.88
AMiD (Ours)	**31.10**	**27.86** ±0.29	**16.46** ±0.41	16.62 ±0.50	**32.64** ±0.26	**35.64** ±0.07	**25.84**