
DISTILLATION OF LARGE LANGUAGE MODELS VIA CONCRETE SCORE MATCHING

Yeongmin Kim¹, Donghyeok Shin¹, Mina Kang¹, Byeonghu Na¹, Il-Chul Moon^{1,2}

¹Korea Advanced Institute of Science and Technology (KAIST), ²summary.ai
{alsdudrla10, tlsehdgur0, kasong13, byeonghu.na, icmoon}@kaist.ac.kr

ABSTRACT

Large language models (LLMs) deliver remarkable performance but are costly to deploy, motivating knowledge distillation (KD) for efficient inference. Existing KD objectives typically match student and teacher probabilities via softmax, which blurs valuable logit information. While direct logit distillation (DLD) mitigates softmax smoothing, it fails to account for logit shift invariance, thereby restricting the solution space. We propose *Concrete Score Distillation* (CSD), a discrete score-matching objective that overcomes both softmax-induced smoothing and restrictions on the optimal solution set. We resolve the training instability and quadratic complexity of discrete score-matching in autoregressive LLMs, and the resulting CSD objective aligns relative logit differences across all vocabulary pairs between student and teacher with flexible weighting. We provide both mode-seeking and mode-covering instances within our framework and evaluate CSD on task-agnostic instruction-following and task-specific distillation using GPT-2-1.5B, OpenLLaMA-7B, and GEMMA-7B-IT. Experiments show that CSD consistently surpasses recent KD objectives, achieves favorable fidelity–diversity trade-offs, and yields complementary gains when combined with on-policy techniques, demonstrating its scalability and effectiveness for LLM distillation.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable generative capabilities across a wide range of tasks (Achiam et al., 2023; Dubey et al., 2024; Liu et al., 2024; Comanici et al., 2025). Such progress has been primarily driven by the vast amount of training data and the unprecedented scale of model parameters (Kaplan et al., 2020). However, when deploying such LLMs in real-world applications, the recurring inference cost becomes prohibitively expensive. Consequently, research into reducing the parameter size of LLMs while preserving performance has become particularly crucial for enabling efficient inference. In this context, knowledge distillation (KD) (Hinton et al., 2015) has emerged as a promising approach for LLMs, as it allows a smaller student model to inherit the capabilities of a large teacher model, thereby enabling more efficient inference.

The common paradigm in KD for LLMs is to align the per-token probability distributions of the student with those of the teacher. Kullback–Leibler (KL) divergence was initially the most widely adopted objective, and the search for more effective probability matching losses has since become a central topic of research. Alternative objectives have been proposed within the framework of f -divergence (Wen et al., 2023; Gu et al., 2024; Agarwal et al., 2024), as well as its smoothed variants (Ko et al., 2024; Shing et al., 2025; Ko et al., 2025). However, existing distillation losses primarily targeted the estimated probabilities obtained through the softmax transformation, instead of directly utilizing the raw neural network outputs (logits) from either the teacher or the student. As illustrated in Figure 1b, even when the teacher’s logit values differ substantially, their corresponding probability values can be nearly indistinguishable. Such smoothing hinders the student from faithfully capturing the teacher’s knowledge, a challenge further exacerbated in modern LLMs with large vocabularies, where most tokens are assigned near-zero probabilities (See Figure 1a).

In traditional KD, direct logit distillation (DLD) (Ba & Caruana, 2014; Urban et al., 2017) has been proposed as an alternative strategy, with advantages in generalization capability and in removing the softmax smoothing (Kim et al., 2021). However, such approaches have not been thoroughly explored

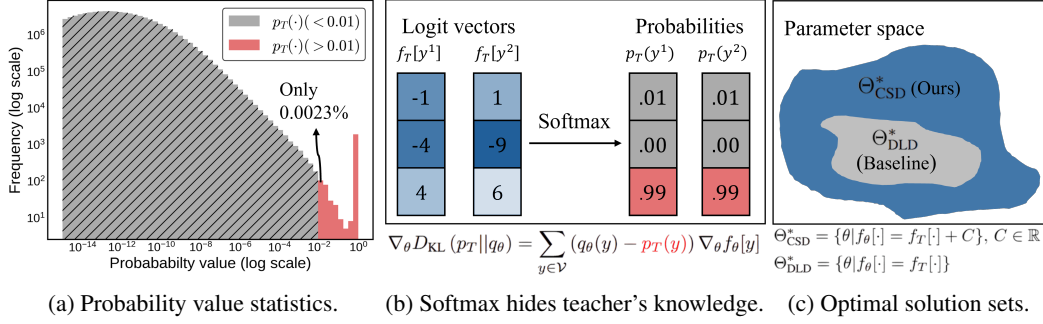


Figure 1: Motivation for logit-level distillation and limitations of prior work. (a) Statistics of per-token probabilities for every vocabulary for 16 input–output sequences from the teacher model (GPT-2-1.5B). The probabilities are highly sparse, with only 0.0023% being greater than 0.01. (b) Despite large differences in logits (e.g., $[-1, -4, 4]$ vs. $[1, -9, 6]$), softmax yields nearly identical probabilities and gradients. (c) Prior direct logit distillation restricts the solution set.

in the context of LLMs. This paper identifies a key drawback of DLD: its restriction on the optimal solution set as described in Figure 1c. Considering the softmax activation in inference, it is sufficient for the teacher’s and student’s logits to agree up to an additive constant, but the previous solutions of DLD fail to accommodate such an acceptable slack constant, a.k.a. *logit shift invariance*. Such a restriction on the solution set may hinder the discovery of optimal solutions in distillation, particularly when the teacher and student models have a large capacity gap, as is often the case with LLMs. Therefore, the goal of this paper is to establish a design space of distillation losses that overcome both the softmax-induced smoothing of teacher knowledge and the restriction on the solution set.

This paper adopts the idea from energy-based models (Song & Kingma, 2021), which design objectives that avoid the constraint of probabilistic models (sum-to-one) by using the score-matching objective (Hyvärinen & Dayan, 2005). We propose *Concrete Score Distillation* (CSD), a discrete form of the score-matching objective (Meng et al., 2022) adapted for autoregressive LLM distillation. We address training instability and computational overhead arising when applying the score-matching objective to LLMs, and provide theoretical guarantees of optimality, showing that its solution set is broader than that of DLD from both theoretical and empirical perspectives. The resulting objective reduces to matching the relative logit differences across all pairs of vocabulary items between the student and teacher, while allowing flexible weighting across all vocabulary pairs in linear time with respect to vocabulary size. Furthermore, we present instances within our framework that exhibit both mode-seeking and mode-covering properties.

In our experiments, we conducted both task-agnostic instruction-following distillation and task-specific distillations (summarization, mathematics, and translation) using GPT-2 (Radford et al., 2019), OpenLLaMA (Geng & Liu, 2023), and GEMMA (Team et al., 2024) backbones. The proposed CSD consistently outperformed recent probability-matching objectives as well as direct logit distillation. By appropriately choosing weighting functions, we further demonstrated that our method resides on the frontier of the diversity–fidelity trade-off. Finally, we observed complementary performance gains when integrating our loss with on-policy techniques.

2 PRELIMINARIES

2.1 KNOWLEDGE DISTILLATION OF LARGE LANGUAGE MODELS

We consider autoregressive large language models (LLMs), consisting of a teacher p_T and a student q_θ , where the student is a smaller and more efficient model. Given an input context $\mathbf{c}$, the student generates an output sequence $\mathbf{y} = (y_1, y_2, \dots, y_L)$ with probability $q_\theta(\mathbf{y}|\mathbf{c}) = \prod_{t=1}^L q_\theta(y_t|\mathbf{c}, \mathbf{y}_{<t})$, where L denotes the sequence length, and the teacher’s probability is defined analogously. Each token y_t is drawn from the fixed vocabulary set $\mathcal{V} := \{v_1, v_2, \dots\}$. As in prior works (Lin et al., 2020; Ko et al., 2024), we assume the teacher and student share the same vocabulary set. To compute the token probability $q_\theta(y_t|\mathbf{c}, \mathbf{y}_{<t})$, an LLM typically adopts a parametric function $f_\theta : \mathcal{V}^{|\mathbf{c}|} \times \mathcal{V}^{t-1} \rightarrow \mathbb{R}^{|\mathcal{V}|}$, which maps the input $(\mathbf{c}, \mathbf{y}_{<t})$ to a logit vector $f_\theta(\mathbf{c}, \mathbf{y}_{<t}) \in \mathbb{R}^{|\mathcal{V}|}$. The logit corresponding to token y_t is denoted by $f_\theta(\mathbf{c}, \mathbf{y}_{<t})[y_t]$. For brevity of notation, the explicit input arguments of the function

f_θ will be omitted hereafter. Let f_T be the parametric function of the teacher. Accordingly, the probability of each token is calculated through the following softmax transformation:

$$q_\theta(y_t|\mathbf{c}, \mathbf{y}_{<t}) = \frac{\exp(f_\theta[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_\theta[x])}, \quad p_T(y_t|\mathbf{c}, \mathbf{y}_{<t}) = \frac{\exp(f_T[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_T[x])}. \quad (1)$$

Problem definition: The goal of knowledge distillation for LLMs is to align the student’s per-token probability distribution with that of the teacher, so that the student inherits the teacher’s capabilities. We assume access to input–output sequence pairs $(\mathbf{c}, \mathbf{y}) \sim \mathcal{D}$, obtained either from a fixed dataset or from samples generated by the student or teacher (Lin et al., 2020; Ko et al., 2024). For each selected instance $(\mathbf{c}, \mathbf{y})$, distillation is performed by selecting a specific discrepancy metric D and minimizing the discrepancy between the per-token probability distributions with respect to θ :

$$\mathbb{E}_{(\mathbf{c}, \mathbf{y}) \sim \mathcal{D}} \left[\frac{1}{L} \sum_{t=1}^L D(p_T(\cdot|\mathbf{c}, \mathbf{y}_{<t}) || q_\theta(\cdot|\mathbf{c}, \mathbf{y}_{<t})) \right]. \quad (2)$$

Prior work and motivation: In previous studies, D is most commonly chosen as the KL divergence (Hinton et al., 2015), which is formulated as follows (the input of the probability is omitted):

$$D_{\text{KL}}(p_T || q_\theta) = \sum_{y_t \in \mathcal{V}} p_T(y_t|\mathbf{c}, \mathbf{y}_{<t}) \log \frac{p_T(y_t|\mathbf{c}, \mathbf{y}_{<t})}{q_\theta(y_t|\mathbf{c}, \mathbf{y}_{<t})}. \quad (3)$$

However, D_{KL} focuses on the teacher’s probabilities and is constrained by the softmax. As shown in Figure 1b, although the teacher carries rich knowledge across all vocabulary items at the logit level, much of it is lost after softmax, and the teacher provides nearly identical gradient signals to most minor tokens. Accordingly, in classical KD studies (Ba & Caruana, 2014; Urban et al., 2017), *direct logit distillation* (DLD) has been widely adopted as a logit-level mean squared error (MSE) loss:

$$\mathcal{L}_{\text{DLD}}(\theta; p_T, w) = \frac{1}{2} \sum_{y_t \in \mathcal{V}} w(y_t) (f_\theta[y_t] - f_T[y_t])^2, \quad (4)$$

where $w(\cdot)$ is a strictly positive weighting function¹. Kim et al. (2021) showed that $\mathcal{L}_{\text{DLD}}$ provides better generalization and representation capability by taking minority indices into account. Since faithfully distilling logit information is crucial for large-vocabulary LLMs, we investigated the use of DLD for LLM distillation. However, we found that its optimal solution does not permit logit constant invariance, thereby severely restricting the solution set. This observation motivated us to develop a logit-level distillation loss that does not restrict the optimal solution.

2.2 SCORE MATCHING FOR A DISCRETE RANDOM VARIABLE

Score-matching (SM) (Hyvärinen & Dayan, 2005) was originally proposed in energy-based models (Song & Kingma, 2021) with continuous variables $\mathbf{x} \in \mathbb{R}^d$. An energy function $E_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$ maps $\mathbf{x}$ to a scalar. The corresponding probability and the score-matching objective are given by:

$$q_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta}, \quad \mathcal{L}_{\text{SM}}(\theta; p_{\text{data}}, w) = \mathbb{E}_{w(\mathbf{x})} [\|\nabla_{\mathbf{x}} \log q_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2], \quad (5)$$

where $Z_\theta = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x})) d\mathbf{x}$ is the partition function, and $w(\cdot)$ is a weighting function. The term $\nabla_{\mathbf{x}} \log q_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} E_\theta(\mathbf{x})$ is known as the *Stein score*, which uniquely identifies the probability distribution without requiring the computation of Z_θ . $\mathcal{L}_{\text{SM}}$ facilitates the design of losses without considering the normalization constraint of probabilistic models. The probability computation q_θ here follows, analogously, the form of the LLM probabilities in Eq. (1). The difference is that an LLM outputs energy values f_θ over all finite states at once, whereas an EBM handles continuous variables, so that each input to E_θ yields only a single scalar output.

Inspired by how EBMs design losses beyond the normalized structure of a probabilistic model through score-matching, we extend this idea to construct logit-level distillation losses for LLMs. However, because the Stein score is defined through derivatives, it cannot be directly applied to discrete random variables. Meng et al. (2022) proposed a generalized score function, applicable to both continuous

¹Throughout this paper, we assume each weighting function sums to one over the vocabulary for simplicity.

and discrete variables, named the *concrete score*: $s_\theta(y) := \left[\frac{q_\theta(x)}{q_\theta(y)} \right]_{x \in \mathcal{V}}$. Similar to the Stein score, the concrete score characterizes local changes at the current state, but replaces them with probability ratios between all other point masses. This term is also uniquely identifiable with the underlying distribution. The corresponding concrete score-matching objective is then defined as:

$$\mathcal{L}_{\text{CSM}}(\theta; p_{\text{data}}, w) = \frac{1}{2} \left[\sum_{y \in \mathcal{V}} \sum_{x \in \mathcal{V}} w(y, x) \left(\frac{q_\theta(x)}{q_\theta(y)} - \frac{p_{\text{data}}(x)}{p_{\text{data}}(y)} \right)^2 \right], \quad (6)$$

where $w(\cdot, \cdot)$ is a positive weighting function. Previous work on language models (Lou et al., 2024) typically adopted this loss by directly parameterizing the concrete score (also known as discrete diffusion models) to mimic the data distribution. In contrast, we take this concept as a starting point to design logit-level distillation losses for autoregressive-type language models, which are more dominant in real-world applications.

3 METHOD

This section introduces the proposed *Concrete Score Distillation* (CSD) objective for knowledge distillation (KD) in autoregressive large language models (LLMs). Section 3.1 discusses the challenges of directly applying $\mathcal{L}_{\text{CSM}}$ to LLMs, so we propose a modified objective with theoretical guarantees of optimality and compare the objective with $\mathcal{L}_{\text{DLD}}$. Section 3.2 presents an efficient analytic gradient computation for CSD, analyzes its gradient structure, and compares it with that of D_{KL} .

3.1 CONCRETE SCORE DISTILLATION FOR LARGE LANGUAGE MODELS

Tackling training instability: We observe that optimizing the student model q_θ by minimizing $\mathcal{L}_{\text{CSM}}(\theta; p_T, w)$ leads to training instability, as the likelihood ratio $\frac{q_\theta(x)}{q_\theta(y_t)}$ can diverge as the denominator approaches zero. In the discrete diffusion model (Lou et al., 2024), a single vocabulary item is fed into the neural network s_θ , which directly outputs the ratios over the other vocabulary items, thereby avoiding instability. In contrast, autoregressive LLMs compute probabilities for each vocabulary item separately and then take their ratios, making this issue specific to autoregressive LLMs.

Training instability is a well-known issue in likelihood ratio estimation (Rhodes et al., 2020). Following Higuchi & Suzuki (2025), we address it by applying a monotonically increasing function to the concrete scores. In particular, we adopt the logarithm, which yields the following objective:

$$\mathcal{L}_{\text{CSD}}(\theta; p_T, w) := \frac{1}{2} \left[\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w(y_t, x) \left(\log \frac{q_\theta(x|\mathbf{c}, \mathbf{y}_{<t})}{q_\theta(y_t|\mathbf{c}, \mathbf{y}_{<t})} - \log \frac{p_T(x|\mathbf{c}, \mathbf{y}_{<t})}{p_T(y_t|\mathbf{c}, \mathbf{y}_{<t})} \right)^2 \right] \quad (7)$$

$$= \frac{1}{2} \sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w(y_t, x) (f_\theta[x] - f_\theta[y_t] - f_T[x] + f_T[y_t])^2. \quad (8)$$

The choice of the logarithm function provides two benefits: (1) it yields an MSE loss between logits (i.e., neural network outputs), ensuring stability by avoiding the likelihood ratio computation; and (2) it naturally leads to the logit-level loss design, which aligns with our motivation.

Logit distillation with intra-vocabulary relationships:

Unlike $\mathcal{L}_{\text{DLD}}$, which directly matches student and teacher logits for the same vocabulary item, $\mathcal{L}_{\text{CSD}}$ aligns the logit residuals across different vocabulary items between the student and the teacher. This allows the student not only to be compared against the teacher but also to perform relative comparisons among its own vocabulary items. In contrast to D_{KL} , where softmax normalization implicitly adjusts each vocabulary item relative to all others, our loss explicitly controls the pairwise relationships between student vocabulary items y_t and x through the weighting function $w(y_t, x)$. Figure 2 illustrates how a logit vector $f_\theta(\mathbf{c}, \mathbf{y}_{<t}) \in \mathbb{R}^{|\mathcal{V}|}$ (e.g., $[1, 4, -3]$) produces a concrete score and how it is matched with the teacher’s concrete score. The following theorems provide the theoretical guarantee of the proposed objective function.

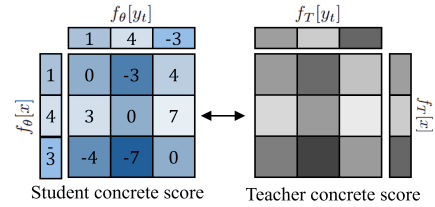


Figure 2: Schematic for $\mathcal{L}_{\text{CSD}}$ (Eq. (8)).

Algorithm 1: Gradient computation of *Concrete Score Distillation*

Input: Student f_θ , teacher f_T , prompt $\mathbf{c}$, prefix $\mathbf{y}_{<t}$, function $w(\cdot, \cdot) = w_1(\cdot)w_2(\cdot)$.

- 1 Compute the student logit $f_\theta[y_t] = f_\theta(\mathbf{c}, \mathbf{y}_{<t}), \forall y_t \in \mathcal{V}$.
 - 2 **with** *no_grad*:
 - 3 Compute the teacher logit $f_T[y_t] = f_T(\mathbf{c}, \mathbf{y}_{<t}), \forall y_t \in \mathcal{V}$.
 - 4 Compute the weighted average logits:
 - 5 $\tilde{f}_\theta^{w_1} = \sum_{y_t \in \mathcal{V}} [w_1(y_t) \times f_\theta[y_t].\text{detach}], \tilde{f}_\theta^{w_2} = \sum_{y_t \in \mathcal{V}} [w_2(y_t) \times f_\theta[y_t].\text{detach}]$
 - 6 $\tilde{f}_T^{w_1} = \sum_{y_t \in \mathcal{V}} [w_1(y_t) \times f_T[y_t]], \tilde{f}_T^{w_2} = \sum_{y_t \in \mathcal{V}} [w_2(y_t) \times f_T[y_t]]$
 - 7 Compute the weighted normalized logits:
 - 8 $\tilde{f}_\theta^{w_1}[y_t] = f_\theta[y_t] - \tilde{f}_\theta^{w_1}, \tilde{f}_\theta^{w_2}[y_t] = f_\theta[y_t] - \tilde{f}_\theta^{w_2}, \forall y_t \in \mathcal{V}$.
 - 9 $\tilde{f}_T^{w_1}[y_t] = f_T[y_t] - \tilde{f}_T^{w_1}, \tilde{f}_T^{w_2}[y_t] = f_T[y_t] - \tilde{f}_T^{w_2}, \forall y_t \in \mathcal{V}$.
 - 10 $w_{\text{grad}}(y_t) = [w_1(y_t) [\tilde{f}_\theta^{w_2}[y_t] - \tilde{f}_T^{w_2}[y_t]] + w_2(y_t) [\tilde{f}_\theta^{w_1}[y_t] - \tilde{f}_T^{w_1}[y_t]]], \forall y_t \in \mathcal{V}$
 - 11 $\nabla_\theta \mathcal{L}_{\text{CSD}}(\theta; p_T, w) = \sum_{y_t \in \mathcal{V}} [w_{\text{grad}}(y_t) \nabla_\theta f_\theta[y_t]]$
 - 12 **return** $\nabla_\theta \mathcal{L}_{\text{CSD}}(\theta; p_T, w)$
-

Proposition 1. (*Consistency*) Given context $\mathbf{c}$ and prefix $\mathbf{y}_{<t}$, assume sufficient model capacity. For any $w(\cdot, \cdot) > 0$, define the set of optimal parameters as $\Theta_{\text{CSD}}^* = \arg \min_\theta \mathcal{L}_{\text{CSD}}(\theta; p_T, w)$. Then, for any $\theta^* \in \Theta_{\text{CSD}}^*$, we have $\mathcal{L}_{\text{CSD}}(\theta^*; p_T, w) = 0$, and the following holds for all $y_t \in \mathcal{V}$:

$$q_{\theta^*}(y_t | \mathbf{c}, \mathbf{y}_{<t}) = p_T(y_t | \mathbf{c}, \mathbf{y}_{<t}).$$

Please refer to Section A.1 for the proof. Proposition 1 shows that consistency holds when matching the log-transformed concrete scores of the student and teacher, and guarantees that our objective leads the student to converge to the target teacher.

Theorem 2. (*Solution Superset*) Under sufficient model capacity, let the set of optimal parameters $\Theta_{\text{CSD}}^* = \arg \min_\theta \mathcal{L}_{\text{CSD}}(\theta; p_T, w)$ and $\Theta_{\text{DLD}}^* = \arg \min_\theta \mathcal{L}_{\text{DLD}}(\theta; p_T, w)$, then following holds:

$$\Theta_{\text{CSD}}^* \supseteq \Theta_{\text{DLD}}^*.$$

Please see Section A.2 for the proof. Theorem 2 implies that all solutions obtainable by $\mathcal{L}_{\text{DLD}}$ can also be recovered by $\mathcal{L}_{\text{CSD}}$. This is because $\mathcal{L}_{\text{CSD}}$ is invariant to constant shifts in logits; for example, when $f_\theta[y_t] = f_T[y_t] + C$ for all $y_t \in \mathcal{V}$, the probabilities are identical and the $\mathcal{L}_{\text{CSD}}$ is zero, whereas the $\mathcal{L}_{\text{DLD}}$ is not optimal. This advantage could be pronounced under limited model capacity, where the larger solution set of $\mathcal{L}_{\text{CSD}}$ enables more faithful approximation of the teacher’s knowledge.

3.2 GRADIENT COMPUTATION AND ANALYSIS

The remaining challenge of the proposed objective $\mathcal{L}_{\text{CSD}}$ in Eq. (8) lies in its computational cost of $\mathcal{O}(|\mathcal{V}|^2)$. Unlike D_{KL} and D_{DLD} , D_{CSD} requires a double summation over the vocabulary set $\mathcal{V}$. This formulation is infeasible to implement in standard computational environments due to memory constraints. Nevertheless, we show that the gradient of this objective can be computed in linear time:

Theorem 3. (*Efficient Gradient Computation*) Assume $w(y_t, x) = w_1(y_t)w_2(x)$, then the gradient of $\mathcal{L}_{\text{CSD}}(\theta; p_T, w)$ with respect to θ could be computed in $\mathcal{O}(|\mathcal{V}|)$ as:

$$\nabla_\theta \mathcal{L}_{\text{CSD}}(\theta; p_T, w) = \sum_{y_t \in \mathcal{V}} \mathbf{w}(y_t)^T \left(\tilde{\mathbf{f}}_\theta[y_t] - \tilde{\mathbf{f}}_T[y_t] \right) \nabla_\theta f_\theta[y_t], \quad (9)$$

where $\mathbf{w}(y_t) = (w_1(y_t), w_2(y_t))^T$, $\tilde{\mathbf{f}}_\theta[y_t] = \left(\tilde{f}_\theta^{w_2}[y_t], \tilde{f}_\theta^{w_1}[y_t] \right)^T$, $\tilde{\mathbf{f}}_T[y_t] = \left(\tilde{f}_T^{w_2}[y_t], \tilde{f}_T^{w_1}[y_t] \right)^T$, with $\tilde{f}_\theta^w[y_t] = f_\theta[y_t] - \mathbb{E}_{w(x)}[f_\theta[x]]$, $\tilde{f}_T^w[y_t] = f_T[y_t] - \mathbb{E}_{w(x)}[f_T[x]]$ are normalized logits.

The proof is provided in Section A.3. These results follow from factorizing the independent variables. Algorithm 1 further details the gradient computation of Eq. (9) step by step, with each step requiring only linear time over the vocabulary. An alternative approach is to use Monte Carlo estimation. Instead of taking a weighted sum over all possible states of y_t with w_1 , one can draw a single sample

of y_t according to probability w_1 and compute the loss in expectation. However, this approach increases the variance within the batched samples, leading to degraded estimation accuracy and typically lower performance compared to the analytic computation (see Figure 5).

Gradient analysis: The gradient of $\mathcal{L}_{\text{CSD}}$ in Eq. (9) has a structure similar to that of D_{KL} . For intuitive understanding, let us consider the case where the weighting function of CSD is the uniform distribution U . Then, the gradient of each loss becomes:

$$\begin{aligned}\nabla_{\theta} D_{\text{KL}}(p_T || q_{\theta}) &= \sum_{y_t \in \mathcal{V}} \left(\underbrace{\frac{\exp(f_{\theta}[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_{\theta}[x])}}_{\text{normalized student logit}} - \underbrace{\frac{\exp(f_T[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_T[x])}}_{\text{normalized teacher logit}} \right) \nabla_{\theta} f_{\theta}[y_t], \\ \nabla_{\theta} \mathcal{L}_{\text{CSD}}(\theta; p_T, U) &= \sum_{y_t \in \mathcal{V}} \frac{2}{|\mathcal{V}|} \left(\underbrace{\left(f_{\theta}[y_t] - \frac{\sum_{x \in \mathcal{V}} f_{\theta}[x]}{|\mathcal{V}|} \right)}_{\text{normalized student logit}} - \underbrace{\left(f_T[y_t] - \frac{\sum_{x \in \mathcal{V}} f_T[x]}{|\mathcal{V}|} \right)}_{\text{normalized teacher logit}} \right) \nabla_{\theta} f_{\theta}[y_t].\end{aligned}$$

In gradient descent, both losses decrease the student’s logit $f_{\theta}[y_t]$ where the student’s normalized logits are large, and increase $f_{\theta}[y_t]$ where the teacher’s normalized logits are large. The only difference lies in how the logit coefficients are normalized over the vocabulary set: D_{KL} inherits the softmax form, which, as noted in Figure 1b, poses a major problem for transferring the teacher’s knowledge. In contrast, our $\mathcal{L}_{\text{CSD}}$ uses centering normalization, allowing the student to directly capture the teacher’s logit information. Moving beyond the uniform weighting case study, the formulation in Eq. (9) further provides a design space for logit normalization through (w_1, w_2) , where w_1 controls the weighting of vocabulary tokens during gradient updates and w_2 governs coefficient normalization, with their roles applied again in reverse order (w_2, w_1) .

4 EXPERIMENTS

This section comprehensively validates the effectiveness of the proposed *Concrete Score Distillation* (CSD) across various experimental setups. Section 4.1 shows results on task-agnostic instruction-following distillation, comparing CSD with alternative loss functions and assessing its performance when combined with recent on-policy methods. Section 4.2 further examines task-specific settings, including math, summarization, and translation, to evaluate the applicability of CSD. Finally, Section 4.3 establishes the contribution of each component in CSD through ablation studies.

4.1 TASK-AGNOSTIC INSTRUCTION-FOLLOWING DISTILLATION

Experimental setup: We follow the training setup of DistiLLM (Ko et al., 2024). For the distillation dataset $\mathcal{D}$, we use databricks-dolly-15k (Conover et al., 2023), containing about 14,000 samples for training, with 500 held out for validation and 500 for evaluation. For comparison with the baseline, we optionally add a pretraining loss using the pretraining dataset OpenWebText (Gokaslan & Cohen, 2019) in some cases of Table 2. We first fine-tune the GPT-2-1.5B (Radford et al., 2019) teacher on the dataset, and then distill it into GPT-2-0.1B and GPT-2-0.3B students. Similarly, we distill OpenLLaMA-7B (Geng & Liu, 2023) into OpenLLaMA-3B. We determined the learning rate and batch size by referring to the search ranges used in prior studies (Gu et al., 2024; Ko et al., 2024). We use the detached student probability as the default choice for both w_1 and w_2 , and analyze alternative choices through ablation studies. Please refer to Section C for further details.

Baselines: Since our main focus is on the loss function, we compared our method with existing objectives using the same teacher checkpoint. The baselines include KL, reverse KL (RKL) (Gu et al., 2024), symmetric KL (the mean of KL and RKL), Jeffrey’s divergence, Total Variation (Wen et al., 2023), Generalized Jensen–Shannon (GJS) (Agarwal et al., 2024) with smoothing parameter 0.9, Skewed KL (SKL) (Ko et al., 2024), Skewed reverse KL (SRKL) (Ko et al., 2024) with smoothing parameter 0.1, and α – β divergence (AB) (Wang et al., 2025) with parameters (0.2, 0.7). We followed the hyperparameter choices reported in each paper and the implementation of DistiLLM. For KL, we performed a full-range hyperparameter search, as in our method. For losses not specified in prior work, we adopted the same settings as for KL.

Evaluation metrics and setups: We evaluated on five instruction-following benchmarks: 1) the test set of Dolly, 2) Self-Instruct (Wang et al., 2023), 3) Vicuna Eval (Chiang et al., 2023), 4) Super-

Table 1: Comparison of loss functions for distilling GPT-2-1.5B into GPT-2-0.1B. Every result is from our implementation with the same teacher, purely using the distillation objective. ROUGE-L scores were averaged over five random seeds; best scores are **boldfaced**, second-best underlined.

Loss	Dolly Eval	Self-Instruct	Vicuna Eval	Super-NI	UnNI	Avg. ($\uparrow$)
Teacher	27.00 $\pm$ 0.19	14.07 $\pm$ 0.37	16.31 $\pm$ 0.32	26.46 $\pm$ 0.41	31.10 $\pm$ 0.06	22.99
KL	23.52 $\pm$ 0.25	10.02 $\pm$ 0.58	14.57 $\pm$ 0.32	16.76 $\pm$ 0.17	18.55 $\pm$ 0.13	16.68
RKL (Gu et al., 2024)	24.26 $\pm$ 0.11	11.19 $\pm$ 0.17	15.80 $\pm$ 0.26	20.17 $\pm$ 0.15	22.99 $\pm$ 0.14	18.88
Sym-KL	23.29 $\pm$ 0.20	10.24 $\pm$ 0.31	15.25 $\pm$ 0.43	17.46 $\pm$ 0.11	20.60 $\pm$ 0.08	17.37
Jeffrey	23.00 $\pm$ 0.38	10.82 $\pm$ 0.44	15.00 $\pm$ 0.50	18.19 $\pm$ 0.11	20.07 $\pm$ 0.11	17.42
TV (Wen et al., 2023)	23.88 $\pm$ 0.30	11.03 $\pm$ 0.51	15.13 $\pm$ 0.44	24.58 $\pm$ 0.25	25.24 $\pm$ 0.06	19.97
GJS (0.9) (Agarwal et al., 2024)	24.10 $\pm$ 0.24	11.40 $\pm$ 0.39	16.02 $\pm$ 0.57	20.28 $\pm$ 0.13	22.55 $\pm$ 0.12	18.87
SKL (0.1) (Ko et al., 2024)	24.17 $\pm$ 0.24	11.21 $\pm$ 0.53	15.29 $\pm$ 0.24	22.65 $\pm$ 0.14	24.69 $\pm$ 0.11	19.60
SRKL (0.1) (Ko et al., 2024)	24.53 $\pm$ 0.21	12.19 $\pm$ 0.29	15.63 $\pm$ 0.22	23.37 $\pm$ 0.27	24.28 $\pm$ 0.18	20.00
AB (0.2, 0.7) (Wang et al., 2025)	24.20 $\pm$ 0.12	11.82 $\pm$ 0.29	15.87 $\pm$ 0.36	21.44 $\pm$ 0.20	25.59 $\pm$ 0.09	19.78
CSD (Ours)	24.94 $\pm$ 0.29	<u>12.06</u> $\pm$ 0.46	15.78 $\pm$ 0.49	24.60 $\pm$ 0.31	25.88 $\pm$ 0.13	20.65

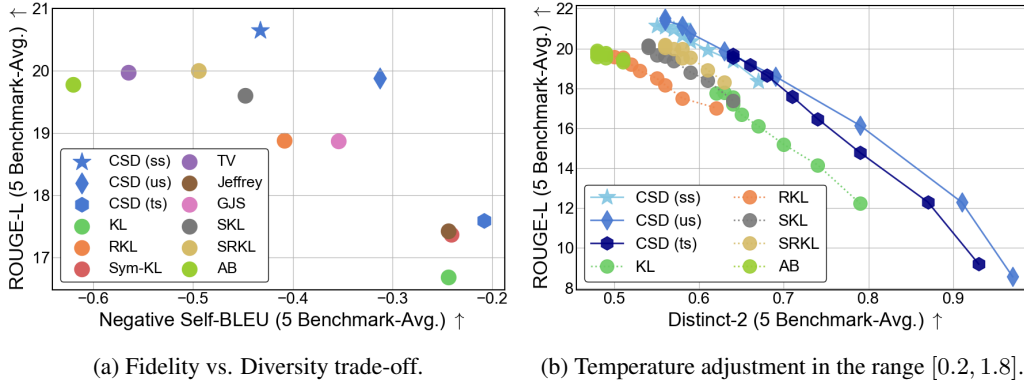


Figure 3: An in-depth analysis of the distributional behavior of different loss functions.

Natural Instructions (Super-NI) (Wang et al., 2022), and 5) Unnatural Instructions (UnNI) (Honovich et al., 2023). ROUGE-L (Lin, 2004), which measures similarity to the golden answer, was used as the primary metric. We additionally employed Self-BLEU (Zhu et al., 2018) and Distinct-N (Li et al., 2016) as diversity metrics. Furthermore, GPT-4 feedback (Zheng et al., 2023) was used as a proxy for human judgment. Checkpoints were saved at each epoch, with evaluation performed on the one achieving the best validation ROUGE-L. The decoding temperature was set to 1 by default, and following prior work, reduced to 0.7 for GPT-judge evaluation.

Loss-level comparison: To purely analyze the effect of the distillation loss itself, this comparison excludes the use of pretraining losses, initialization with an SFT-tuned student, and any on-policy techniques. Table 1 shows that the proposed CSD objective outperforms the other nine objectives, ranking first on three of the five benchmarks, second on one, and achieving the highest average score. SKL (Ko et al., 2024) and AB (Wang et al., 2025) exhibit slightly lower performance than previously reported, likely due to their reliance on pretraining losses or on-policy techniques.

Figure 3a shows the fidelity–diversity trade-off based on ROUGE-L and Self-BLEU scores. Traditionally, KL favors diversity, whereas RKL favors mode-seeking. Within this trade-off, SKL, SRKL, TV, and AB achieve higher ROUGE-L scores than RKL, but at the cost of reduced diversity, reflecting a stronger emphasis on fidelity. Diversity, however, remains an important aspect of user experience in instruction-following, and it becomes a valuable metric as it enhances performance when combined with best-of-N sampling. The proposed CSD provides an additional lever to control the fidelity–diversity trade-off. By default, using the detached student probabilities (S, S) yields the highest fidelity. Replacing one side with uniform (U, S) or with the teacher (T, S) gradually increases diversity. This is likely because the (S, S) makes the model focus only on regions where the student already assigns a high likelihood, limiting its exploratory ability. The trade-off offered by CSD envelopes those of existing losses, and we expect that even better operating points may exist within the design space of w_1 and w_2 . Figure 3b presents an ablation on temperature, which enables easy adjustment of the trade-off during inference. Even within a reasonable range of decoding temperature,

Table 2: Instruction-following performance of CSD with on-policy techniques for various backbones. $\mathcal{D}$ denotes the distillation dataset. ROUGE-L scores are averaged over five random seeds, with the best score for each student highlighted in **bold**.

Method	Loss	$\mathcal{D}$	Dolly Eval	Self-Instruct	Vicuna Eval	Super-NI	UnNI	Avg. ($\uparrow$)
Teacher (GPT-2-1.5B)			27.00 $\pm$ 0.19	14.07 $\pm$ 0.37	16.31 $\pm$ 0.32	26.46 $\pm$ 0.41	31.10 $\pm$ 0.06	22.99
Teacher (OpenLLaMA-7B)			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
GPT-2-1.5B $\rightarrow$ GPT-2-0.1B								
GKD (Agarwal et al., 2024)	GJS	Mix	22.48 $\pm$ 0.20	10.08 $\pm$ 0.67	15.61 $\pm$ 0.08	13.88 $\pm$ 0.21	16.59 $\pm$ 0.13	15.73
DistiLLM (Ko et al., 2024)	SKL	Ada	25.28 $\pm$ 0.28	12.04 $\pm$ 0.49	16.66 $\pm$ 0.34	22.13 $\pm$ 0.31	24.32 $\pm$ 0.14	20.09
ImitKD (Lin et al., 2020)	KL	On	21.79 $\pm$ 0.18	10.25 $\pm$ 0.37	14.65 $\pm$ 0.62	17.35 $\pm$ 0.12	19.43 $\pm$ 0.13	16.69
GKD + Ours	CSD	Mix	25.50 $\pm$ 0.34	12.03 $\pm$ 0.65	16.65 $\pm$ 0.45	21.39 $\pm$ 0.14	23.48 $\pm$ 0.03	19.81
DistiLLM + Ours	CSD	Ada	25.34 $\pm$ 0.27	11.93 $\pm$ 0.36	16.99 $\pm$ 0.29	22.96 $\pm$ 0.24	24.72 $\pm$ 0.09	20.39
ImitKD + Ours	CSD	On	25.70 $\pm$ 0.23	12.40 $\pm$ 0.48	17.18 $\pm$ 0.52	22.91 $\pm$ 0.46	25.47 $\pm$ 0.17	20.73
GPT-2-1.5B $\rightarrow$ GPT-2-0.3B								
GKD (Agarwal et al., 2024)	GJS	Mix	25.15 $\pm$ 0.41	11.22 $\pm$ 0.33	16.45 $\pm$ 0.48	17.35 $\pm$ 0.29	22.25 $\pm$ 0.05	18.48
DistiLLM (Ko et al., 2024)	SRKL	Ada	26.92 $\pm$ 0.23	13.75 $\pm$ 0.29	16.90 $\pm$ 0.25	26.12 $\pm$ 0.27	29.65 $\pm$ 0.14	22.67
ImitKD (Lin et al., 2020)	KL	On	23.61 $\pm$ 0.34	12.37 $\pm$ 0.26	15.53 $\pm$ 0.27	20.20 $\pm$ 0.20	24.42 $\pm$ 0.29	19.23
GKD + Ours	CSD	Mix	27.11 $\pm$ 0.42	13.71 $\pm$ 0.45	16.98 $\pm$ 0.29	25.49 $\pm$ 0.35	30.16 $\pm$ 0.13	22.69
DistiLLM + Ours	CSD	Ada	26.77 $\pm$ 0.18	13.96 $\pm$ 0.62	17.05 $\pm$ 0.34	26.29 $\pm$ 0.08	29.56 $\pm$ 0.09	22.72
ImitKD + Ours	CSD	On	27.14 $\pm$ 0.28	14.85 $\pm$ 0.66	16.88 $\pm$ 0.18	26.28 $\pm$ 0.21	30.43 $\pm$ 0.04	23.12
OpenLLaMA-7B $\rightarrow$ OpenLLaMA-3B								
TAID (Shing et al., 2025)	tKL	Ada	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 (Ko et al., 2024)	SKL	Ada	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 (Ko et al., 2024)	SRKL	Ada	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
ImitKD + Ours	CSD	On	29.63 $\pm$ 0.40	21.81 $\pm$ 0.47	20.37 $\pm$ 0.51	36.49 $\pm$ 0.13	36.86 $\pm$ 0.10	29.03

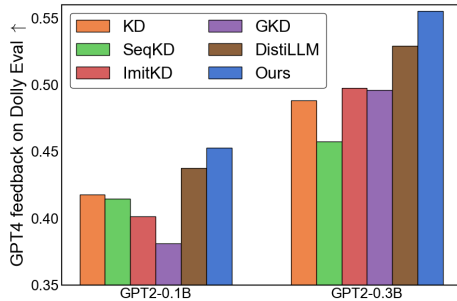


Figure 4: GPT-4 feedback performance, showing the proportion of responses judged correct relative to the golden answers. The teacher’s score is 0.61.

CSD achieves better trade-off points than other losses. In particular, CSD (U, S) demonstrates a well-balanced exchange between diversity and fidelity through temperature adjustment.

Orthogonal improvement with recent on-policy advances: Table 2 reports the performance of recent distillation baselines augmented with the CSD loss, demonstrating its orthogonal applicability. We applied the CSD (S, S) loss to ImitKD (Lin et al., 2020), GKD (Agarwal et al., 2024), and DistiLLM (Ko et al., 2024). The primary distinction among these methods, apart from their losses, lies in the choice of dataset $\mathcal{D}$: ImitKD uses purely student-generated on-policy data, GKD combines fixed data with student outputs, and DistiLLM adaptively selects between them based on validation loss. As a result, the average ROUGE-L score improved for both GPT-2-0.1B and GPT-2-0.3B students across all cases. The best result on each benchmark was also achieved by our method, with particularly strong performance under pure on-policy settings. We also evaluated using GPT-4 as the judge in Figure 4, where our best model was judged superior to other baselines. There may exist CSD variants other than (S, S) that perform better for specific $\mathcal{D}$, but we leave this exploration to future work. Finally, applying our best setting to a larger OpenLLaMA also outperformed recent baselines, demonstrating the scalability of CSD with respect to model size. We also provide comparisons with more baselines in Table 5 of Section D.

Table 3: Task-specific distillation performance from the GEMMA-7B-IT teacher to the GEMMA-2B-IT student.

Loss	Summarization	Translation	GSM8K
	ROUGE-L	COMET	Accuracy
Teacher	37.09	79.23	60.27
KL	35.02	73.96	24.03
JS	35.60	74.05	23.73
TV	27.49	73.73	0.00
Jeffrey	35.29	74.02	23.28
SKL	25.86	59.65	0.00
SRKL	26.68	73.10	0.00
CSD (Ours)	35.67	74.14	25.78

Table 4: Ablation on the logit-level loss design space. T , U , and S denote teacher, uniform, and detached student probabilities. ROUGE-L scores are averaged over five seeds; best scores are in bold.

Loss	$w_1(\cdot)$	$w_2(\cdot)$	Dolly Eval	Self-Instruct	Vicuna Eval	Super-NI	UnNI	Avg. ($\uparrow$)
DLD	T	-	0.09	0.07	0.18	0.07	0.06	0.09
	U	-	11.25	5.55	9.10	9.02	8.24	8.63
	S	-	24.22	12.01	15.42	25.44	24.88	20.39
CSD (Ours)	T	T	6.82	4.24	9.16	4.53	4.83	5.91
	U	U	17.21	8.08	14.27	13.19	14.07	13.37
	S	S	24.94	12.06	15.78	24.60	25.88	20.65
	U	S	24.15	12.25	15.25	22.55	25.19	19.88
	T	S	22.77	10.62	14.06	18.81	21.71	17.59

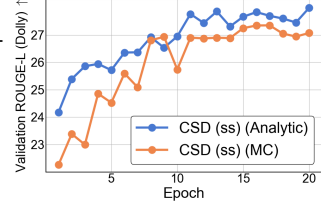


Figure 5: Ablation between analytic gradient calculation (Eq. (9)) and Monte Carlo sampling for the (S, S) weighting $\mathcal{L}_{\text{CSD}}$ calculation.

4.2 TASK-SPECIFIC DISTILLATION

Experimental setup: We verify the effectiveness of CSD across diverse tasks, including dialogue summarization, low-resource translation, and arithmetic reasoning. Experiments were conducted on the DialogSum (Chen et al., 2021), Flores-200 (Costa-Jussà et al., 2022), and GSM8K (Cobbe et al., 2021) datasets. We used GEMMA-7B-IT (Team et al., 2024), fine-tuned with SFT as the teacher and GEMMA-2B-IT as the student, following the experimental setup of Xu et al. (2025). We compared with the baselines using the same teacher, changing only the loss function. Please see Section C for further details.

Results: Table 3 presents a performance comparison across three tasks against the baseline loss functions. The proposed CSD objective function achieved the best performance on all three tasks under identical experimental conditions. In the arithmetic reasoning task, we observed multiple cases where certain losses resulted in zero accuracy. A case study in Tables 7 to 10 in Section D shows that these models often generate excessively long reasoning steps without reaching a final answer, reflecting a failure to learn proper formatting; moreover, much of the reasoning itself was incorrect. As illustrated in Figure 3a, the losses TV, SKL, and SRKL exhibit mode-seeking tendencies, which we conjecture may have caused collapses into suboptimal modes depending on the task and model. In contrast, CSD obtained stable performance by applying (T, S) weighting in this case.

4.3 ABLATION STUDIES

Table 4 presents ablation studies on the weighting function choices in CSD and compares them with direct logit distillation (DLD). DLD accommodates only a single weighting function. When comparing DLD with T , U , and S against CSD with (T, T) , (U, U) , and (S, S) , respectively, our method consistently achieved higher average scores. As shown in Theorem 2, we hypothesize that the broader solution space positively contributed to this improvement. Figure 8 in Section D shows that DLD restricts solutions to those with a residual constant of zero, CSD adapts residual constants per token, providing evidence that it explores a broader solution set. Beyond performance, CSD provides a more flexible loss design space through two weighting functions. As illustrated in Figure 3a, replacing (S, S) with (U, S) or (T, S) reduces ROUGE-L but increases diversity, highlighting that CSD can adapt to tasks requiring either mode-covering or mode-seeking properties. Figure 5 compares the method to resolve $\mathcal{O}(|\mathcal{V}|^2)$ computational cost: 1) using analytic gradient computation from Theorem 3 and 2) Monte Carlo sampling. Employing the analytic form led to faster training and improved convergence, demonstrating that the value of Theorem 3 translates into empirical performance.

5 CONCLUSION

We introduced *Concrete Score Distillation* (CSD), a novel design space for distillation losses in large language models. CSD simultaneously addresses the challenges of softmax-induced smoothing and restrictions on the optimal solution set, which prior methods have failed to resolve together. Within this framework, we presented instances of both mode-covering and mode-seeking, and demonstrated scalability by consistently surpassing prior work across diverse tasks and model backbones up to 7B parameters. We anticipate that even better instances can be discovered within the proposed design space, particularly by refining w_1 and w_2 and adapting them to the type of data (fixed or on-policy). This points to promising directions for future exploration of improved instances.

REFERENCES

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, 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. URL <https://openreview.net/forum?id=3zKtaqxLhW>.
- Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.
- 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/>.
- 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.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- 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.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.
- 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.
- 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>.
- Rei Higuchi and Taiji Suzuki. Direct density ratio optimization: A statistically consistent approach to aligning large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=jDdaysR6bz>.
- 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/>.
- Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Taehyeon Kim, Jaehoon Oh, Nak Yil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.
- Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL <https://aclanthology.org/D16-1139/>.
- Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/ko24c.html>.
- 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. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=rc65N9xIrY>.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, 2016.
- Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.494. URL <https://aclanthology.org/2020.emnlp-main.494/>.
- 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/>.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, pp. 32819–32848. PMLR, 2024.

-
- Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.
- 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.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névélol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri (eds.), *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 578–585, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.wmt-1.52/>.
- Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.
- Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. TAID: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=cqsw28DuMW>.
- Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.
- 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.
- Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Abdelrahman Mohamed, Matthai Philipose, Matt Richardson, and Rich Caruana. Do deep convolutional nets really need to be deep and convolutional? In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=r10FA8Kxg>.
- 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. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=vt65VjJakt>.
- 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/>.

-
- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. 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. 10817–10834, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.605. URL <https://aclanthology.org/2023.acl-long.605/>.
- Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler divergence in knowledge distillation for large language models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL <https://aclanthology.org/2025.coling-main.383/>.
- Wenda Xu, Rujun Han, Zifeng Wang, Long 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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=EgJhwYR2tB>.
- 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. Texus: 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 PROOFS AND DERIVATIONS

A.1 PROOF OF PROPOSITION 1

Proposition 1. (*Consistency*) Given context $\mathbf{c}$ and prefix $\mathbf{y}_{<t}$, assume sufficient model capacity. For any $w(\cdot, \cdot) > 0$, define the set of optimal parameters as $\Theta_{CSD}^* = \arg \min_{\theta} \mathcal{L}_{CSD}(\theta; p_T, w)$. Then, for any $\theta^* \in \Theta_{CSD}^*$, we have $\mathcal{L}_{CSD}(\theta^*; p_T, w) = 0$, and the following holds for all $y_t \in \mathcal{V}$:

$$q_{\theta^*}(y_t | \mathbf{c}, \mathbf{y}_{<t}) = p_T(y_t | \mathbf{c}, \mathbf{y}_{<t}).$$

Proof. We have the following objective:

$$\mathcal{L}_{CSD}(\theta; p_T, w) := \frac{1}{2} \left[\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w(y_t, x) \left(\log \frac{q_{\theta}(x | \mathbf{c}, \mathbf{y}_{<t})}{q_{\theta}(y_t | \mathbf{c}, \mathbf{y}_{<t})} - \log \frac{p_T(x | \mathbf{c}, \mathbf{y}_{<t})}{p_T(y_t | \mathbf{c}, \mathbf{y}_{<t})} \right)^2 \right] \quad (10)$$

$$= \frac{1}{2} \sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w(y_t, x) (f_{\theta}[x] - f_{\theta}[y_t] - f_T[x] + f_T[y_t])^2. \quad (11)$$

Since the objective is a weighted sum of squares with strictly positive weights $w(\cdot, \cdot) > 0$, the loss attains its minimum if and only if each squared term vanishes, i.e.

$$f_{\theta^*}[x] - f_{\theta^*}[y_t] = f_T[x] - f_T[y_t], \quad \forall y_t, x \in \mathcal{V}. \quad (12)$$

Then, the probability of a student satisfying the following:

$$q_{\theta^*}(y_t | \mathbf{c}, \mathbf{y}_{<t}) = \frac{\exp(f_{\theta^*}[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_{\theta^*}[x])} = \frac{\exp(f_{\theta^*}[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_{\theta^*}[y_t] + f_T[x] - f_T[y_t])} \quad (13)$$

$$= \frac{\exp(f_T[y_t])}{\sum_{x \in \mathcal{V}} \exp(f_T[x])} = p_T(y_t | \mathbf{c}, \mathbf{y}_{<t}). \quad (14)$$

□

A.2 PROOF OF THEOREM 2

Theorem 2. (*Solution Superset*) Under sufficient model capacity, let the set of optimal parameters $\Theta_{CSD}^* = \arg \min_{\theta} \mathcal{L}_{CSD}(\theta; p_T, w)$ and $\Theta_{DLD}^* = \arg \min_{\theta} \mathcal{L}_{DLD}(\theta; p_T, w)$, then following holds:

$$\Theta_{CSD}^* \supsetneq \Theta_{DLD}^*.$$

Proof. We have the following objective for direct logit distillation (DLD):

$$\mathcal{L}_{DLD}(\theta; p_T, w) = \frac{1}{2} \sum_{y_t \in \mathcal{V}} w(y_t) (f_{\theta}[y_t] - f_T[y_t])^2, \quad (15)$$

Since the loss is expressed as a strictly positive weighted sum of squares, it achieves its minimum value only when all squared terms are individually zero, i.e.,

$$f_{\theta_{DLD}^*}[y_t] = f_T[y_t], \quad \forall y_t \in \mathcal{V}. \quad (16)$$

Unlike DLD, the optimality condition of our loss is more relaxed. Specifically, it is sufficient for θ^* to satisfy the condition in Eq. (12), i.e.,

$$f_{\theta^*}[y_t] = f_T[y_t] + C, \quad \forall y_t \in \mathcal{V}, \quad C \in \mathbb{R}. \quad (17)$$

At $C = 0$, our objective recovers the solution set of DLD; for an arbitrary choice of C , it yields a strictly larger optimal solution set. This arises from the fact that the softmax mapping used to express probabilities is invariant under additive constants, whereas DLD explicitly constrains this constant to coincide with that of the teacher, which consequently reduces the solution set.

□

A.3 PROOF OF THEOREM 3

Theorem 3. (Efficient Gradient Computation) Assume $w(y_t, x) = w_1(y_t)w_2(x)$, then the gradient of $\mathcal{L}_{\text{CSD}}(\theta; p_T, w)$ with respect to θ could be computed in $\mathcal{O}(|\mathcal{V}|)$ as:

$$\nabla_{\theta} \mathcal{L}_{\text{CSD}}(\theta; p_T, w) = \sum_{y_t \in \mathcal{V}} \mathbf{w}(y_t)^T \left(\tilde{\mathbf{f}}_{\theta}[y_t] - \tilde{\mathbf{f}}_T[y_t] \right) \nabla_{\theta} f_{\theta}[y_t], \quad (9)$$

where $\mathbf{w}(y_t) = (w_1(y_t), w_2(y_t))^T$, $\tilde{\mathbf{f}}_{\theta}[y_t] = \left(\tilde{f}_{\theta}^{w_2}[y_t], \tilde{f}_{\theta}^{w_1}[y_t] \right)^T$, $\tilde{\mathbf{f}}_T[y_t] = \left(\tilde{f}_T^{w_2}[y_t], \tilde{f}_T^{w_1}[y_t] \right)^T$, with $\tilde{f}_{\theta}^w[y_t] = f_{\theta}[y_t] - \mathbb{E}_{w(x)}[f_{\theta}[x]]$, $\tilde{f}_T^w[y_t] = f_T[y_t] - \mathbb{E}_{w(x)}[f_T[x]]$ are normalized logits.

Proof. We have the following objective:

$$\mathcal{L}_{\text{CSD}}(\theta; p_T, w) = \frac{1}{2} \sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) \left(\log \frac{q_{\theta}(x|\mathbf{c}, \mathbf{y}_{<t})}{q_{\theta}(y_t|\mathbf{c}, \mathbf{y}_{<t})} - \log \frac{p_T(x|\mathbf{c}, \mathbf{y}_{<t})}{p_T(y_t|\mathbf{c}, \mathbf{y}_{<t})} \right)^2 \quad (18)$$

$$= \frac{1}{2} \sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (f_{\theta}[x] - f_{\theta}[y_t] - f_T[x] + f_T[y_t])^2. \quad (19)$$

And its gradient is given by:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{CSD}}(\theta; p_T, w) &= \sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (f_{\theta}[x] - f_{\theta}[y_t] - f_T[x] + f_T[y_t]) \nabla_{\theta} (f_{\theta}[x] - f_{\theta}[y_t]) \\ &= \underbrace{\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (f_{\theta}[x] - f_T[x]) \nabla_{\theta} (f_{\theta}[x])}_{\textcircled{1}} - \underbrace{\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (f_{\theta}[x] - f_T[x]) \nabla_{\theta} (f_{\theta}[y_t])}_{\textcircled{2}} \\ &+ \underbrace{\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (-f_{\theta}[y_t] + f_T[y_t]) \nabla_{\theta} (f_{\theta}[x])}_{\textcircled{3}} - \underbrace{\sum_{y_t \in \mathcal{V}} \sum_{x \in \mathcal{V}} w_1(y_t)w_2(x) (-f_{\theta}[y_t] + f_T[y_t]) \nabla_{\theta} (f_{\theta}[y_t])}_{\textcircled{4}} \end{aligned}$$

$$\textcircled{1} = \sum_{y_t \in \mathcal{V}} w_1(y_t) \times \sum_{x \in \mathcal{V}} w_2(x) (f_{\theta}[x] - f_T[x]) \nabla_{\theta} (f_{\theta}[x]) = \sum_{y_t \in \mathcal{V}} w_2(y_t) (f_{\theta}[y_t] - f_T[y_t]) \nabla_{\theta} (f_{\theta}[y_t])$$

$$\textcircled{4} = \sum_{x \in \mathcal{V}} w_2(x) \times \sum_{y_t \in \mathcal{V}} w_1(y_t) (f_{\theta}[y_t] - f_T[y_t]) \nabla_{\theta} (f_{\theta}[y_t]) = \sum_{y_t \in \mathcal{V}} w_1(y_t) (f_{\theta}[y_t] - f_T[y_t]) \nabla_{\theta} (f_{\theta}[y_t])$$

$$\begin{aligned} \textcircled{2} &= - \left\{ \sum_{x \in \mathcal{V}} w_2(x) (f_{\theta}[x] - f_T[x]) \right\} \times \left\{ \sum_{y_t \in \mathcal{V}} w_1(y_t) \nabla_{\theta} (f_{\theta}[y_t]) \right\} \\ &= - \mathbb{E}_{w_2(x)} [f_{\theta}[x] - f_T[x]] \times \sum_{y_t \in \mathcal{V}} w_1(y_t) \nabla_{\theta} (f_{\theta}[y_t]) \end{aligned}$$

$$\begin{aligned} \textcircled{3} &= - \left\{ \sum_{y_t \in \mathcal{V}} w_1(y_t) (f_{\theta}[y_t] - f_T[y_t]) \right\} \times \left\{ \sum_{x \in \mathcal{V}} w_2(x) \nabla_{\theta} (f_{\theta}[x]) \right\} \\ &= - \mathbb{E}_{w_1(y_t)} [f_{\theta}[y_t] - f_T[y_t]] \times \sum_{x \in \mathcal{V}} w_2(x) \nabla_{\theta} (f_{\theta}[x]) \\ &= - \mathbb{E}_{w_1(x)} [f_{\theta}[x] - f_T[x]] \times \sum_{y_t \in \mathcal{V}} w_2(y_t) \nabla_{\theta} (f_{\theta}[y_t]) \end{aligned}$$

$$\begin{aligned}
\textcircled{4} + \textcircled{2} &= \sum_{y_t \in \mathcal{V}} w_1(y_t) (f_\theta[y_t] - f_T[y_t] - \mathbb{E}_{w_2(x)}[f_\theta[x] - f_T[x]]) \nabla_\theta(f_\theta[y_t]) \\
&= \sum_{y_t \in \mathcal{V}} w_1(y_t) (\tilde{f}_\theta^{w_2}[y_t] - \tilde{f}_T^{w_2}[y_t]) \nabla_\theta(f_\theta[y_t]) \\
\textcircled{1} + \textcircled{3} &= \sum_{y_t \in \mathcal{V}} w_2(y_t) (f_\theta[y_t] - f_T[y_t] - \mathbb{E}_{w_1(x)}[f_\theta[x] - f_T[x]]) \nabla_\theta(f_\theta[y_t]) \\
&= \sum_{y_t \in \mathcal{V}} w_2(y_t) (\tilde{f}_\theta^{w_1}[y_t] - \tilde{f}_T^{w_1}[y_t]) \nabla_\theta(f_\theta[y_t]) \\
\textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} &= \sum_{y_t \in \mathcal{V}} \mathbf{w}(y_t)^T (\tilde{\mathbf{f}}_\theta[y_t] - \tilde{\mathbf{f}}_T[y_t]) \nabla_\theta(f_\theta[y_t])
\end{aligned}$$

□

B RELATED WORKS

The choice of discrepancy metric between teacher and student probability distributions is central to knowledge distillation for large language models (LLMs). Prior work has predominantly employed either forward KL divergence (Hinton et al., 2015) or reverse KL divergence (Gu et al., 2024). These divergences, however, exhibit distinct biases: forward KL is inherently mode-covering, while reverse KL is mode-seeking. Consequently, optimization under either measure imposes an unavoidable trade-off between fidelity and diversity. To address this limitation, recent studies have explored alternative measures, including (generalized) Jensen–Shannon divergence (Wen et al., 2023; Agarwal et al., 2024), adaptive KL divergence (Wu et al., 2025), and α – β divergence (Wang et al., 2025). Complementarily, Ko et al. (2024) introduced skew KL and skew reverse KL divergences to improve optimization stability. Beyond the KL family, total variation distance has also been investigated (Wen et al., 2023). Broadly, existing approaches extend in two directions: (i) instantiating different generating functions within the f -divergence family, or (ii) constructing hybrid objectives that combine multiple divergences. In contrast, we propose a novel logit-level distillation framework grounded in concrete-score matching (Meng et al., 2022), which departs from the f -divergence family and offers both extensibility and originality. Furthermore, we introduce a loss design space with multiple instances, including instances that envelope the diversity–fidelity trade-off exhibited by previous methods.

Concurrently, a complementary line of work has examined dataset composition to mitigate the distribution mismatch between training and inference. Several studies have explored on-policy strategies, either using only student-generated outputs (Lin et al., 2020) or combining them with a fixed dataset (Agarwal et al., 2024) and teacher-generated outputs (Gu et al., 2024; Xu et al., 2025). To reduce the computational overhead of on-policy training, Ko et al. (2024) proposed an adaptive off-policy method with a replay buffer. By contrast, our contribution focuses on developing a novel discrepancy metric, which is orthogonal to these dataset composition strategies and can be seamlessly integrated with them as shown in Table 2.

C EXPERIMENTAL DETAILS

Task-agnostic instruction-following distillation: All experiments were conducted primarily on four RTX 3090 GPUs. Following prior work (Gu et al., 2024; Ko et al., 2024; Wang et al., 2025), we searched learning rates in [5e-4, 1e-4, 5e-5] and batch sizes in [8, 16, 32]. Each configuration was trained for 20 epochs, saving a checkpoint at every epoch, and evaluated using the checkpoint with the highest validation ROUGE-L score. We used the same five evaluation seeds [10, 20, 30, 40, 50] as in prior work to compute the mean and standard deviation of the evaluation metric. The baselines in Table 2 were run with the official code settings of prior work (Ko et al., 2024), with additional tuning for the batch size. In the OpenLLaMA experiments, all baselines and ours were standardized to a batch size of 8, the maximum supported in our environment. Baselines used the learning rates from their official code, while we fixed the learning rate to 1e-4 (commonly effective for GPT-2) with CSD,

without further tuning. For ablation studies in Table 4 and Figure 5, we used the same configuration: learning rate 1e-4 and batch size 8. For GPT-4 feedback in Figure 4, we use the following templates following prior work (Zheng et al., 2023; Ko et al., 2024) as shown below. We computed the ratio between the model answer and the golden answer for each of the 500 samples from Dolly Eval, and reported the average over all samples. We provide the reference implementation for CSD in Code 1.

Task-specific distillation: We follow the experimental setup of Xu et al. (2025) with a fixed dataset. For teacher SFT, we trained summarization and arithmetic reasoning for 3 epochs and translation for 10 epochs, using the full datasets. Model evaluation was performed every 16 steps, and the checkpoint with the lowest validation loss was selected. The batch size was fixed to 128 for all tasks, with the learning rate set to 1e-5. For each task in distillation, we distilled both the baselines and our method from the same teacher checkpoint with a fixed learning rate of 1e-5 and batch size of 8, using about 1,000 samples. We trained for 3 epochs on summarization and arithmetic reasoning, and 10 epochs on translation. For the baselines, checkpoints were saved every 25 steps, and the one with the lowest validation loss was used for evaluation. For CSD, since the loss itself cannot be directly computed and training relies on its gradient, validation loss was unavailable; thus, we evaluated using the final checkpoint. For all tasks, we set w_1 and w_2 using the teacher’s and student’s probabilities. For evaluation, we used task-specific metrics: COMET (Rei et al., 2022) for translation, ROUGE-L (Lin, 2004) for summarization, and answer accuracy for arithmetic reasoning, all evaluated on each task’s test dataset.

GPT-4 feedback template

[System] Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: “[[rating]]”, for example: “Rating: [[5]]”.

[Question]
{question}

[The Start of Assistant’s Answer]
{answer}
[The End of Assistant’s Answer]

```
import torch
import torch.nn.functional as F

def CSD_loss(student_logits, teacher_logits, mode):
    student_probs = F.softmax(student_logits, dim=-1)
    teacher_probs = F.softmax(teacher_logits, dim=-1)

    if mode == "SS":
        loss = (student_logits - teacher_logits - torch.sum(student_probs * (student_logits -
            teacher_logits), dim=-1, keepdim=True)).detach() * student_probs.detach() *
            student_logits

    elif mode == "TS":
        loss1 = (student_logits - teacher_logits - torch.sum(teacher_probs * (student_logits -
            teacher_logits), dim=-1, keepdim=True)).detach() * student_probs.detach() *
            student_logits
        loss2 = (student_logits - teacher_logits - torch.sum(student_probs * (student_logits -
            teacher_logits), dim=-1, keepdim=True)).detach() * teacher_probs * student_logits
        loss = (loss1 + loss2) / 2

    distil_loss = torch.sum(loss, dim=-1) ## summation over vocab

    return distil_loss
```

Code 1: CSD loss function implementation

D ADDITIONAL EXPERIMENTAL RESULTS

This section presents additional experimental results. Figure 6 shows the logit and probability statistics of the GPT-2-1.5B teacher, corresponding to Figure 1. Figure 7 illustrates further fidelity–diversity trade-offs using Distinct-N metrics, corresponding to Figure 3a. Figure 8 demonstrates that DLD converges only to solutions with zero residual constants, whereas CSD learns token-dependent residual constants, leading to better convergence points (as evidenced by ROUGE-L scores). Figure 9 presents validation ROUGE-L scores during training, corresponding to Table 1. CSD not only converges to a higher point but also achieves faster performance gains in the early stages. Table 5 provides comparisons with additional baselines corresponding to Table 2, and Table 6 compares CSD with the MSE probability-matching objective under different weighting schemes. Finally, Tables 7 to 10 present case studies of model generations for math questions.

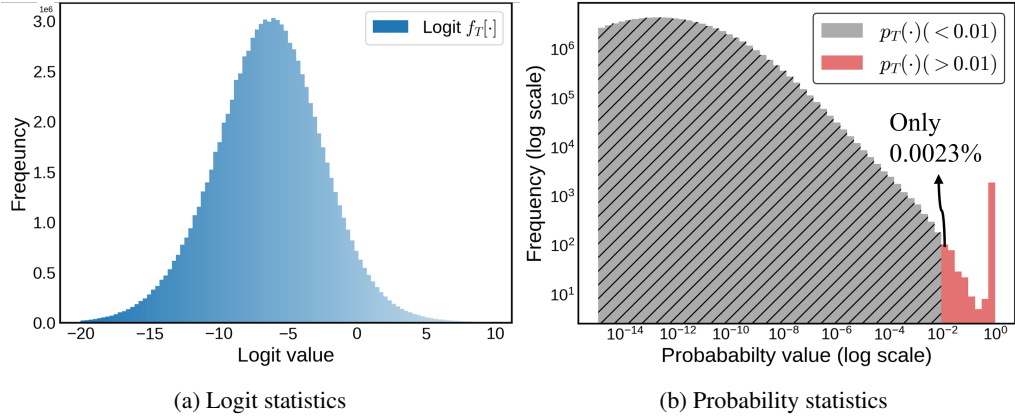


Figure 6: Comparison between teacher’s logit and probability statistics. While the logits span a wide range from -20 to 5 and convey rich information, the probabilities are mostly concentrated near zero.

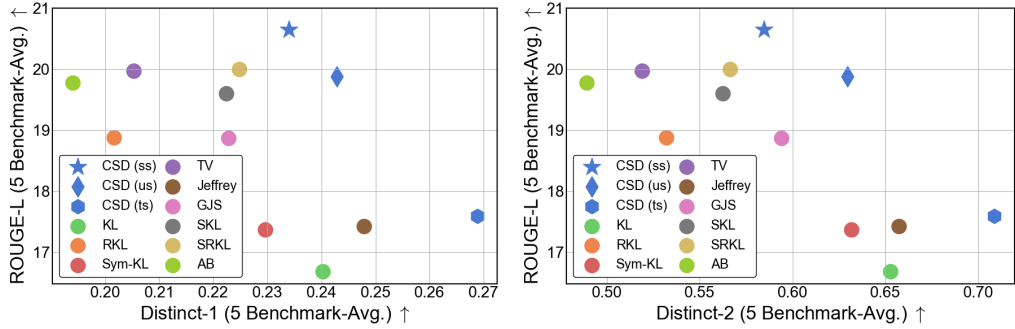


Figure 7: Fidelity vs. Diversity trade-off with more metrics.

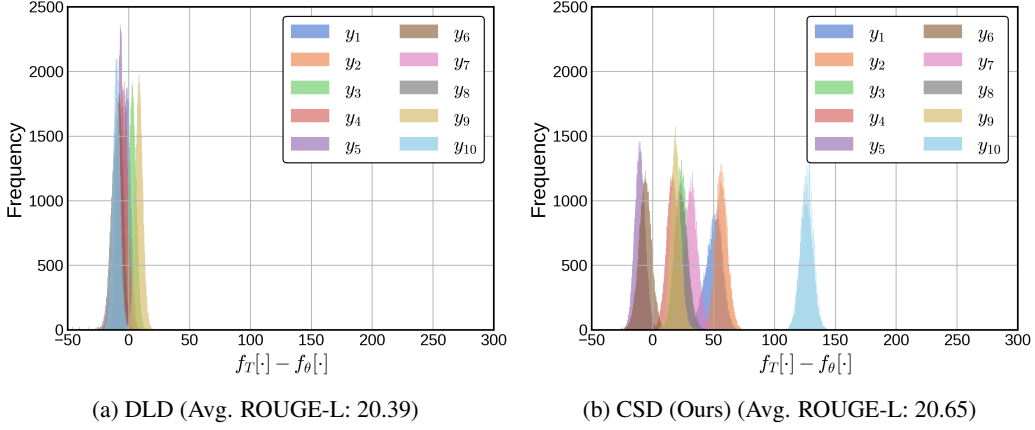


Figure 8: Solution set restriction of direct logit distillation (DLD) and the flexible selection of logit residual constants in *Concrete Score Distillation* (CSD). CSD achieves better instruction-following performance with broader solution sets.

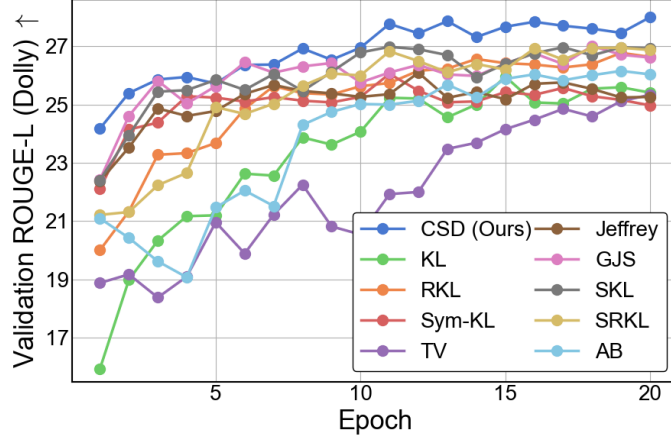


Figure 9: Validation ROUGE-L scores over training epochs.

E THE USE OF LARGE LANGUAGE MODELS

In this work, LLMs were used only for minor writing assistance, such as grammar correction after drafting. In addition, since the research topic is LLM distillation, LLMs were employed as the subject of experiments and also as evaluation models for performance assessment.

Table 5: Comparison with more baselines corresponds to Table 2.

Method	Loss	$\mathcal{D}$	Dolly Eval	Self-Instruct	Vicuna Eval	Super-NI	UnNI	Avg. ($\uparrow$)
Teacher (GPT-2-1.5B)			27.00 $\pm$ 0.19	14.07 $\pm$ 0.37	16.31 $\pm$ 0.32	26.46 $\pm$ 0.41	31.10 $\pm$ 0.06	22.99
GPT-2-1.5B $\rightarrow$ GPT-2-0.1B								
SFT	SFT	Fix	23.49 $\pm$ 0.25	10.56 $\pm$ 0.29	15.09 $\pm$ 0.48	17.13 $\pm$ 0.12	19.97 $\pm$ 0.08	17.25
SeqKD (Kim & Rush, 2016)	SFT	p_T	23.86 $\pm$ 0.49	11.67 $\pm$ 0.80	14.73 $\pm$ 0.37	21.04 $\pm$ 0.19	23.55 $\pm$ 0.11	18.97
KD (Hinton et al., 2015)	KL	Fix	23.52 $\pm$ 0.25	10.02 $\pm$ 0.58	14.57 $\pm$ 0.32	16.76 $\pm$ 0.17	18.55 $\pm$ 0.13	16.68
Ours	CSD	Fix	24.94 $\pm$ 0.29	12.06 $\pm$ 0.46	15.78 $\pm$ 0.49	24.60 $\pm$ 0.31	25.88 $\pm$ 0.13	20.65
Ours	CSD	On	25.70 $\pm$ 0.23	12.40 $\pm$ 0.48	17.18 $\pm$ 0.52	22.91 $\pm$ 0.46	25.47 $\pm$ 0.17	20.73
GPT-2-1.5B $\rightarrow$ GPT-2-0.3B								
SFT	SFT	Fix	25.09 $\pm$ 0.62	12.23 $\pm$ 0.79	16.24 $\pm$ 0.40	23.42 $\pm$ 0.11	26.99 $\pm$ 0.13	20.79
SeqKD (Kim & Rush, 2016)	SFT	p_T	24.79 $\pm$ 0.26	11.03 $\pm$ 0.95	15.27 $\pm$ 0.30	18.91 $\pm$ 0.29	21.78 $\pm$ 0.10	18.36
KD (Hinton et al., 2015)	KL	Fix	25.41 $\pm$ 0.52	11.15 $\pm$ 0.20	15.83 $\pm$ 0.26	20.13 $\pm$ 0.38	23.57 $\pm$ 0.13	19.22
Ours	CSD	On	27.14 $\pm$ 0.28	14.85 $\pm$ 0.66	16.88 $\pm$ 0.18	26.28 $\pm$ 0.21	30.43 $\pm$ 0.04	23.12

Table 6: Comparison with MSE loss between probabilities corresponds to Table 4.

Loss	$w_1(\cdot)$	$w_2(\cdot)$	Dolly Eval	Self-Instruct	Vicuna Eval	Super-NI	UnNI	Avg. ($\uparrow$)
Prob L2	T	-	24.41	11.45	14.43	24.08	25.53	19.98
	U	-	16.43	6.51	9.73	10.94	13.16	11.35
	S	-	15.62	6.59	10.63	10.31	12.51	11.13
CSD (Ours)	T	T	6.82	4.24	9.16	4.53	4.83	5.91
	U	U	17.21	8.08	14.27	13.19	14.07	13.37
	S	S	24.94	12.06	15.78	24.60	25.88	20.65
	U	S	24.15	12.25	15.25	22.55	25.19	19.88
	T	S	22.77	10.62	14.06	18.81	21.71	17.59

Table 7: Qualitative comparison on the GSM8K dataset. Only **CSD (Ours)** produces the correct final answer; other students give incorrect results.

Prompt for model:

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? A:

Correct Answer:

It takes $2/2 = \langle 2/2 = 1 \rangle 1$ bolt of white fiber

So the total amount of fabric is $2 + 1 = \langle 2 + 1 = 3 \rangle 3$ bolts of fabric

3

Teacher sampled output:

The robe takes $2 \times 1/2 = \langle 2 * 1/2 = 1 \rangle 1$ bolt of white fiber.

So it takes $2 + 1 = \langle 2 + 1 = 3 \rangle 3$ bolts in total.

3

CSD (ours) student sampled output:

The robe takes $2 \times 1/2 = \langle 2 * 1/2 = 1 \rangle 1$ bolt of white fiber.

So, it takes $2 + 1 = \langle 2 + 1 = 3 \rangle 3$ bolts in total.

3 (✓)

KL student sampled output:

2 bolts of blue fiber + 1 bolt of white fiber = $\langle 2 + 1 = 3 \rangle 3$ bolts in total

3 bolts of blue fiber + 2 bolts of white fiber = $\langle 3 + 2 = 5 \rangle 5$ bolts in total

5 (✗)

Jeffrey student sampled output:

The robe takes $2 \times 2 = \langle 2 * 2 = 4 \rangle 4$ bolts of blue fiber.

The robe takes $4 / 2 = \langle 4 / 2 = 2 \rangle 2$ bolts of white fiber.

The robe takes $4 + 2 = \langle 4 + 2 = 6 \rangle 6$ bolts in total.

6 (✗)

JS student sampled_output:

The robe takes $2 * 2 = \langle 2 * 2 = 4 \rangle 4$ bolts of blue fiber.

The robe takes $4 / 2 = \langle 4 / 2 = 2 \rangle 2$ bolts of white fiber.

The robe takes $4 + 2 = \langle 4 + 2 = 6 \rangle 6$ bolts in total.

6 (✗)

Table 8: Qualitative comparison on a GSM8K dataset. The SRKL student’s sampled output shows degenerative repetition and drifting arithmetic reasoning.

Prompt for model:

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? A:

Correct Answer:

It takes $2/2=\ll 2/2=1\gg 1$ bolt of white fiber

So the total amount of fabric is $2+1=\ll 2+1=3\gg 3$ bolts of fabric

3

SRKL student sampled output:

The robe takes $2/2=1/2$ bolts of white fiber.

So, it takes $1/2*2=1$ bolt of blue fiber.

So, it takes $1+2=3$ bolts in total.

So, it takes $3*1=3$ bolts in total.

So, it takes $3*2=6$ bolts in total.

So, it takes $6+1=7$ bolts in total.

So, it takes $7*2=14$ bolts in total.

So, it takes $14+3=17$ bolts in total.

So, it takes $17*2=34$ bolts in total.

So, it takes $34+1=35$ bolts in total.

So, it takes 35 bolts in total.

So, it takes $35*2=70$ bolts in total.

So, it takes $70+1=71$ bolts in total.

So, it takes 71 bolts in total.

So, it takes $71*2=142$ bolts in total.

So, it takes $142+3=145$ bolts in total.

So, it takes $145*2=290$ bolts in total.

So, it takes $290+1=291$ bolts in total.

So, it takes 291 bolts in total.

So, it takes $291*2=582$ bolts in total.

So, it takes 582 bolts in total.

So, it takes $582*2=1164$ bolts in total.

So, it takes 1164 bolts in total.

So, it takes $1164*2=2328$ bolts in total.

So, it takes 2328 bolts in total.

So, it takes $2328*2=4656$ bolts in total.

So, it takes 4656 bolts in total.

So, it takes $4656*2=9312$ bolts in total.

So, it takes 9312 bolts in total.

So, it takes $9312*2=18624$ bolts in total.

So, it takes 18624 bolts

Table 9: Qualitative comparison on a GSM8K dataset. The TV student’s sampled output shows degenerative repetition and drifting arithmetic reasoning.

Prompt for model:

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? A:

Correct Answer:

It takes $2/2 = \langle 2/2 = 1 \rangle 1$ bolt of white fiber

So the total amount of fabric is $2+1 = \langle 2+1 = 3 \rangle 3$ bolts of fabric

3

TV student sampled output:

The robe takes $2 \times .5 = 1$ bolt of white fiber.

So it takes $2 + 1 = 3$ bolts in total.

The robe takes $3 \times 2 = 6$ bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

... (repeats) ...

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

The robe takes 6 bolts in total.

So it takes 6 bolts in total.

Table 10: Qualitative comparison on a GSM8K dataset. The SKL student’s sampled output shows degenerative repetition and drifting arithmetic reasoning.

Prompt for model:

Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? A:

Correct Answer:

It takes $2/2=\ll 2/2=1 \gg 1$ bolt of white fiber

So the total amount of fabric is $2+1=\ll 2+1=3 \gg 3$ bolts of fabric

3

SKL student sampled output:

The robe takes $2/2=1/2$ bolts of white fiber.

So it takes $2+1/2=3/2$ bolts of fiber in total.

So it takes $3/2=1.5$ bolts of fiber in total.

So it takes $1.5*2=3$ bolts of fiber in total.

So it takes $3+1=4$ bolts of fiber in total.

So it takes $4*2=8$ bolts of fiber in total.

So it takes $8+1=9$ bolts of fiber in total.

So it takes $9+1=10$ bolts of fiber in total.

So it takes $10+2=12$ bolts of fiber in total.

So it takes $12+1=13$ bolts of fiber in total.

So it takes $13+1=14$ bolts of fiber in total.

So it takes $14+4=18$ bolts of fiber in total.

So it takes $18+1=19$ bolts of fiber in total.

So it takes $19+1=20$ bolts of fiber in total.

So it takes $20+2=22$ bolts of fiber in total.

So it takes $22+1=23$ bolts of fiber in total.

So it takes $23+1=24$ bolts of fiber in total.

So it takes $24+1=25$ bolts of fiber in total.

So it takes $25+1=26$ bolts of fiber in total.

So it takes $26+1=27$ bolts of fiber in total.

So it takes $27+1=28$ bolts of fiber in total.

So it takes $28+1=29$ bolts of fiber in total.

So it takes $29+1=30$ bolts of fiber in total.

So it takes $30+1=31$ bolts of fiber in total.

So it takes $31+1=32$ bolts of fiber in total.

So it takes $32+1=33$ bolts of fiber in total.

So it takes $33+1=34$ bolts of fiber in total.

So it takes $34+1=35$ bolts of fiber in total.

So it takes $35+1=36$ bolts of fiber in total.
