Knowledge Distillation with Perturbed Loss: From a Vanilla Teacher to a Proxy Teacher

Rongzhi Zhang*
Georgia Institute of Technology
Atlanta, GA, USA
rongzhi.zhang@gatech.edu

Jiaming Shen†
Google
New York City, NY, USA
jmshen@google.com

Tianqi Liu†
Google
New York City, NY, USA
tianqiliu@google.com

Jialu Liu
Google
New York City, NY, USA
jialu@google.com

Michael Bendersky†
Google
Mountain View, CA, USA
bemike@google.com

Marc Najork†
Google
Mountain View, CA, USA
najork@google.com

Chao Zhang
Georgia Institute of Technology
Atlanta, GA, USA
chaozhang@gatech.edu

ABSTRACT

Knowledge distillation is a popular technique to transfer knowledge from a large teacher model to a small student model. Typically, the student learns to imitate the teacher by minimizing the KL divergence of its output distribution with the teacher's output distribution. In this work, we argue that such a learning objective is sub-optimal because there exists a discrepancy between the teacher's output distribution and the ground truth label distribution. Therefore, forcing the student to blindly imitate the unreliable teacher output distribution leads to inferior performance. To this end, we propose a novel knowledge distillation objective *PTLoss* by first representing the vanilla KL-based distillation loss function via a Maclaurin series and then *perturbing* the leading-order terms in this series. This perturbed loss implicitly transforms the original teacher into a *proxy teacher* with a distribution closer to the ground truth distribution. We establish the theoretical connection between this "distribution closeness" and the student model generalizability, which enables us to select the PTLoss's perturbation coefficients in a principled way. Extensive experiments on six public benchmark datasets demonstrate the effectiveness of PTLoss with teachers of different scales.

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**.

KEYWORDS

Knowledge Distillation, Large Language Models

---

*This work was conducted during an internship at Google Research.
†Now with Google DeepMind.

1 INTRODUCTION

Knowledge distillation (KD) is a widely-used technique to transfer knowledge from a large teacher model into a much smaller student model with minimum sacrifice of teacher model's predictive power [4, 11]. The vanilla training objective in KD such as KL loss [11, 22, 28] encourages the student's outputs to be close to the teacher's outputs as much as possible. This objective implicitly assumes the teacher's output distribution on the distillation data is perfectly aligned with the ground truth distribution. However, in many applications, this assumption does not hold and the teacher's output distributions can be biased from the ground truth due to various factors, such as the inductive bias encoded in the teacher model architecture [34], miscalibration in the training procedure [22], or the bias in the teacher model training set [20, 21]. Enforcing the student to blindly imitate the teacher's outputs can make the student inherit such biases and produce suboptimal predictions.

To overcome this challenge, one common approach [11] suggests scaling the teacher's logits via a temperature parameter. A proper temperature value can enhance the quality of the teacher model's output distribution by making it closer to the true label distribution [22]. However, the shifting space offered by temperature scaling is limited, and the optimal temperature value relies on resource-intensive grid search. Along a separate line, label smoothing [29] is proposed to regularize the neural networks, and modulated loss functions [18, 19] are designed to address various statistical issues in model training such as overfitting and data imbalance. Despite their potential, there is a lack of work that explores tailoring such techniques for more robust knowledge distillation.
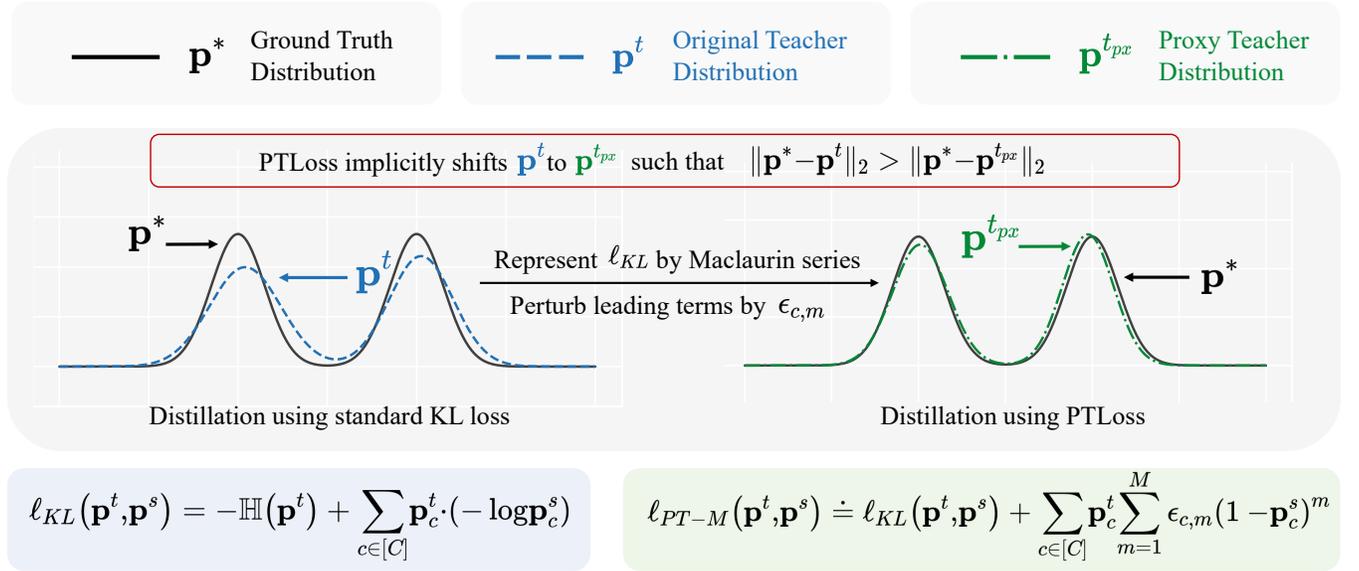
Figure 1: PTLoss implicitly transforms the original teacher into a proxy teacher with a distribution closer to the ground truth distribution. This approach addresses the issue of sub-optimal student models resulting from discrepancies between the teacher's output distribution and the ground truth distribution. By introducing perturbation to standard KL loss represented by its Maclaurin series, we obtain a better proxy teacher, which leads to a more effectively distilled student.

In this study, we propose *PTLoss* for knowledge distillation, which generalizes the vanilla KL loss function and implicitly creates a debiased teacher distribution closer to the ground truth. As shown in Figure 1, our approach does not merely mimic the original teacher model's output distribution, denoted as $\mathbf{p}^t$, but rather aims to bring the distilled model closer to the ground truth distribution $\mathbf{p}^*$, by generating a proxy teacher distribution $\mathbf{p}^{t_{px}}$. This is achieved by the proposed PTLoss, which implicitly transforms the original teacher (the dashed blue curve in Figure 1) into a proxy teacher (the dash-dot green curve in Figure 1). The resultant proxy teacher distribution is closer to the ground truth than the original teacher's output distribution, thus enhancing the distillation process.

Compared to the standard KL loss, PTLoss refrains from forcing an out-and-out imitation of the original teacher model. Instead, PTLoss moderates the distillation objective by adding perturbations to the standard KL loss. Specifically, we first decompose the standard KL loss into its Maclaurin series and then perturb its leading-order terms to construct a more flexible learning objective. This manipulation enables consequential adjustments to the teacher's output distribution. The perturbations are not arbitrary but are meticulously calculated. To determine the perturbation extent, we compute the equivalent distribution of this implicitly shifted teacher's output distribution after perturbations (named "*proxy teacher*") and measure the empirical deviation between the proxy teacher and the ground truth data. It leads to a systematic searching strategy for the perturbation coefficients — the near-optimal perturbation coefficients should minimize the deviation between the distillation risk and the population risk on the validation set.

Theoretically, we justify the effectiveness of PTLoss by proving that it can reduce the deviation from the distillation risk compared to KL loss. We draw a connection between the PTLoss and other perturbation methods (*e.g.,* temperature scaling [11], label smoothing [29], and focal loss [19]). We illustrate that the PTLoss can debias the teacher to produce higher-fidelity outputs via a finer-grained perturbation, while subsuming existing perturbation techniques as special cases. Experiments on multiple datasets with different-sized teacher models demonstrate the empirical advantages of PTLoss.

In summary, we make the following contributions:

(1) A new knowledge distillation loss function PTLoss, which formulates the vanilla KD loss in the form of Maclaurin series and perturbs it to improve the fidelity of teacher models;
(2) A principled method to compute the proxy teacher for determining the perturbation coefficients in PTLoss;
(3) Theoretical analysis on why PTLoss can lower the distillation risk bound and how it subsumes other perturbation methods;
(4) Comprehensive experiments on multiple language understanding tasks with different-sized teacher models showing the advantage of PTLoss.

2 PRELIMINARIES

Multi-class Classification. In a multi-class classification problem with C classes, we are given a set of training examples $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ where input $x_n \in X$ and output y_n is a one-hot vector in $Y = \{y | y \in \{0, 1\}^C, \mathbf{1}^T y = 1\}$ indicating the target label of example x_n. The goal is to learn a probability predictor $\mathbf{p} : X \to \mathbb{R}^C$ by optimizing the below minimal *risk*:

$$R(\mathbf{p}) = \mathbb{E}_{(x,y)}[\ell(y, \mathbf{p}(x))]. \tag{1}$$

where $\ell(y, \mathbf{p}(x))$ is the loss of predicting $\mathbf{p}(x)$ when the true label of example x is y.

A canonical loss function is the cross-entropy loss: $\ell_{CE}(y, \mathbf{p}(x)) = -y\log(\mathbf{p}(x))$ and we may further approximate the above risk via the *empirical risk* on the training set $\mathcal{D}$:

$$\hat{R}(\mathbf{p}; \mathcal{D}) \doteq \frac{1}{N}\sum_{n=1}^{N} y_n(-\log(\mathbf{p}(x_n))) \qquad (2)$$

Our Problem Formulation. In this work, we study the knowledge distillation problem where the labeled training set $\mathcal{D}$ is inaccessible[1]. Specifically, we are only given an unlabeled distillation set $\mathcal{D}_u$, a teacher model $\mathbf{p}^t$, and asked to learn a student model $\mathbf{p}^s$.

Standard Distillation Strategy. A standard KD strategy [11] is to replace the ground truth one-hot label y_n in Eq. 2 with the teacher model's output probabilistic label estimate $\mathbf{p}^t(x_n)$ and utilize the KL divergence loss to learn the student model $\mathbf{p}^s$ via the *distillation empirical risk*:

$$\tilde{R}_{KL}(\mathbf{p}^s; \mathbf{p}^t, \mathcal{D}_u) \doteq \frac{1}{N_u}\sum_{n=1}^{N_u} \ell_{KL}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right), \qquad (3)$$

where $N_u = |\mathcal{D}_u|$ and $\ell_{KL}(\mathbf{p}, \mathbf{q}) = KL(\mathbf{p}||\mathbf{q}) = \mathbf{p}^T\log(\mathbf{p}) - \mathbf{p}^T\log(\mathbf{q})$.

3 PERTURBED DISTILLATION LOSS

Using the KL divergence loss (in short "KL loss") for distillation essentially assumes the teacher model is perfect and forces the student model to mimic the teacher's output label distribution. In reality, the teacher model can produce a biased estimate of label distribution and lead to a sub-optimal student model, as demonstrated by both theoretical analysis [22] and empirical observations [23] (as well as our experiments in Section 5.1).

In this work, we present a new distillation loss that generalizes the standard KL loss to accommodate various degrees of distribution gaps between the biased teacher's output distribution and the underlying ground truth distribution. Inspired by the PolyLoss [18], we propose to first replace the logarithmic terms in the standard KL loss with their corresponding Maclaurin series and then perturb the polynomial terms as follows:

$$\log(x) = -\sum_{m=1}^{\infty} \frac{(1-x)^m}{m} \xrightarrow[\text{term coefficients}]{\text{Perturb polynomial}} \qquad (4)$$

$$\log(x) \approx -\sum_{m=1}^{\infty}\left(\frac{1}{m} + \epsilon_m\right)(1-x)^m.$$

Here, we essentially replace the original coefficient $\frac{1}{m}$ of the m-th order polynomial term in the standard KL loss to $(\frac{1}{m} + \epsilon_m)$. By further replacing the logarithmic terms in standard KL loss (Eq. 3) with the above Eq. 4, we will have:

$$\ell_{KL}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right) = -\mathbb{H}\left(\mathbf{p}^t(x_n)\right) + \sum_{c\in[C]} \mathbf{p}_c^t(x_n)\left[-\log\mathbf{p}_c^s(x_n)\right]$$

$$\approx -\mathbb{H}\left(\mathbf{p}^t(x_n)\right) + \sum_{c\in[C]} \mathbf{p}_c^t(x_n)\left[-\log\mathbf{p}_c^s(x_n) + \sum_{m=1}^{\infty}\epsilon_{c,m}(1-\mathbf{p}_c^s(x_n))^m\right], \qquad (5)$$

---

[1]This setting reflects the real-world scenario where large teacher models (e.g., ChatGPT [24] and GPT4 [25]) only expose their outputs and/or APIs without original training data because of their large model sizes and cautions toward data leakage/misuse.

where $\mathbf{p}_c^t(x_n)$ and $\mathbf{p}_c^s(x_n)$ denote the probability that example x_n belongs to the class c according to the teacher (student) model, and $\mathbb{H}\left(\mathbf{p}^t(x_n)\right)$ is the entropy of the teacher output distribution.

We can further separate out the perturbation coefficients on the right hand side of Eq. 5 and merge $\sum_{c\in[C]} \mathbf{p}_c^t(x_n)\left[-\log\mathbf{p}_c^s(x_n)\right]$ with $\mathbb{H}\left(\mathbf{p}^t(x_n)\right)$ to obtain our perturbed distillation loss:

$$\ell_{PT}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right) \doteq \ell_{KL}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right)$$

$$+ \sum_{c\in[C]} \mathbf{p}_c^t(x_n)\sum_{m=1}^{\infty}\epsilon_{c,m}\left(1-\mathbf{p}_c^s(x_n)\right)^m. \qquad (6)$$

The above equation presents our perturbed distillation loss in its most general form. In practice, however, we cannot tune infinite number of coefficients $\epsilon_{c,m}$ and thus we propose to only tune the first M leading polynomial coefficients while keeping the rest unchanged as follows:

$$\ell_{PT-M}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right) \doteq \ell_{KL}\left(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)\right)$$

$$+ \sum_{c\in[C]} \mathbf{p}_c^t(x_n)\sum_{m=1}^{M}\epsilon_{c,m}\left(1-\mathbf{p}_c^s(x_n)\right)^m. \qquad (7)$$

We can see that if we set all $\epsilon_{c,m}$ to 0, the ℓ_{PT} falls back to the ℓ_{KL} and thus the perturbed distillation loss can be considered as a generalization of the standard KL loss.

Figure 2 presents how PTLoss adjusts biased teachers. For visualization simplicity, we set the number of classes $C = 2$. In Figure 2a, we vary the teacher probability to show how the biased teacher model will impact the distilled student model under either the standard KL loss or our proposed PTLoss. We observe that PTLoss can guide the student's predictions toward the ground truth and thus effectively reduces the inherent bias in the teacher's output probabilities. In Figure 2b, we demonstrate PTLoss enables a diverse shift space to the loss curve. By setting the perturbation coefficients, PTLoss allows flexible adjustments to the loss curve. Combining with our perturbation coefficients selection methods discussed in Sec. 4.3, we can determine the perturbation to optimize the distillation process.

### 3.1 Connections to other perturbation methods

We establish the connections between PTLoss and other related methods that transform the teacher output probabilities, such as label smoothing [29], temperature scaling [11], and focal loss [19]. We show that the loss shift space produced by PTLoss encompasses these techniques and thus PTLoss can offer additional adjustment capabilities. We present the connections in this section and detail the derivation in Appendix 8.3.

KL Loss. The connection between PTLoss and the standard KL loss is quite direct. As we represent the standard KL loss in Maclaurin series and add perturbations, we can easily revert PTLoss to the standard KL loss by setting all perturbation coefficients $\epsilon_{c,m}$ to 0.

Focal Loss. Focal loss incorporates a factor $(1-p)^\gamma$ in the loss function. We demonstrate PTLoss can subsume focal loss by expressing the perturbation coefficients as a function of the factor $(1-p)^\gamma$. For simplicity, we denote $\mathbf{p}^t(x_n)$ and $\mathbf{p}^s(x_n)$ as $\mathbf{p}^t$ and $\mathbf{p}^s$.
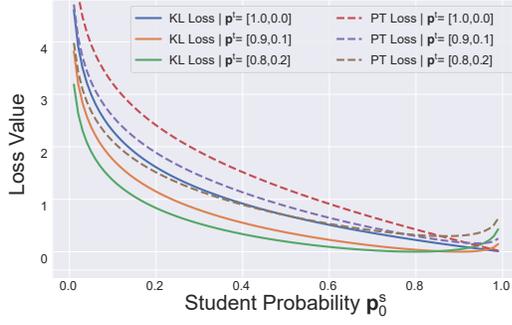
(a) **Loss values with different teacher probabilities** $\mathbf{p}_0^t$ **in** $\{1.0, 0.9, 0.8\}$. **For PTLoss, we fix the perturbation order as 1 and the perturbation coefficients** $\epsilon = [1, 1]$. **Consider the case that ground truth probability is** $[1, 0]$, **PTLoss adjusts the student's predictions by nudging them towards the ground truth, effectively mitigating the bias present in the teacher's output probabilities.**
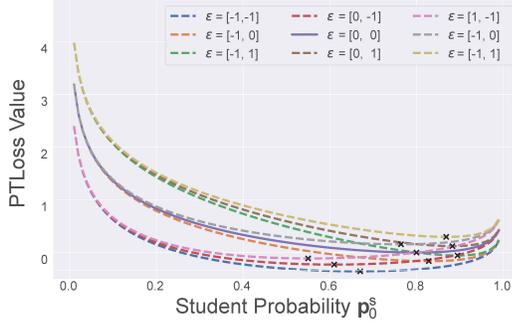
(b) **PTLoss values with different perturbations. We fix the teacher probability** $\mathbf{p}_0^t = [0.8, 0.2]$ **and vary the perturbation coefficients** ϵ, **while the perturbation order is always fixed to** 1. **The black cross denotes the best student model output** $\mathbf{p}_0^s$ **that achieves the lowest loss value. This shows PTLoss can enable flexible adjustments to the loss curve and effectively reduces the bias of the teacher's output.**

Figure 2: Intuitive understanding of our PTLoss for the binary classification task.

By applying Focal loss to KD, we have:

$$\ell_{focal}\left(\mathbf{p}^t, \mathbf{p}^s\right) = -\mathbb{H}\left(\mathbf{p}^t\right) + \sum_{c \in [C]} \mathbf{p}_c^t (1 - \mathbf{p}_c^s)^\gamma [-\log \mathbf{p}_c^s], \quad (8)$$

where $(1 - \mathbf{p}_c^s)^\gamma$ is a factor and the parameter $\gamma > 0$ reduces the relative loss for well-classified examples.

To bridge the connection between PTLoss and the focal loss, we compare Eq. 8 to Eq. 5 and establish the following relationship:

$$\sum_{m=1}^{\infty} \left(\frac{1}{m} + \epsilon_{c,m}\right)(1 - \mathbf{p}_c^s)^m = \sum_{m=1}^{\infty} \frac{(1 - \mathbf{p}_c^s)^\gamma}{m} \cdot (1 - \mathbf{p}_c^s)^m, \quad (9)$$

which leads to the perturbation coefficients as follows:

$$\epsilon_{c,m} = \frac{(1 - \mathbf{p}_c^s)^\gamma - 1}{m}. \quad (10)$$

By incorporating the derived perturbation coefficients $\epsilon_{c,m}$ in our proposed method, we demonstrate that PTLoss can effectively subsume the focal loss. In other words, PTLoss generalizes the focal loss and can adapt to more modulating factors $(1 - p)^\gamma$ for handling the class imbalance problem and improving the knowledge distillation performance.

Temperature Scaling. We compare PTLoss with temperature scaling and claim that PTLoss subsumes it with appropriate approximation. As described in Hinton et al. [11], the logits are adjusted by a temperature to control sharpness or smoothness of the probability distribution:

$$\mathbf{p}_{\tau,0} = \frac{exp(z_0/\tau)}{exp(z_0/\tau) + exp(z_1/\tau)}, \quad (11)$$

where τ is the temperature and z_c is the logits. Here we use binary classification (i.e., $c = 0, 1$) for the derivation simplicity without loss of generality.

We denote $\mathbf{p}_{\tau,c}^t, \mathbf{p}_{\tau,c}^s$ as the teacher, student probability scaled by temperature τ. Incorporating temperature scaling, the KL loss can be formulated as:

$$\ell_{KL}^{temp}\left(\mathbf{p}_\tau^t, \mathbf{p}_\tau^s\right) = -\mathbb{H}\left(\mathbf{p}_\tau^t\right) + \sum_{c \in [C]} \mathbf{p}_{\tau,c}^t \cdot (-\log \mathbf{p}_{\tau,c}^s). \quad (12)$$

Similarly, to bridge PTLoss with temperature scaling, we establish equality between Eq. 12 and Eq. 5, this leads to

$$\epsilon_{c,m} = \frac{a}{m} \cdot \left(1 + \left(\frac{1 - b}{1 - \mathbf{p}_c^s}\right)^m\right) - \frac{1}{m}, \quad (13)$$

where $a = exp(z_1^t - z_0^t)/exp(\frac{z_1^t - z_0^t}{\tau}), b = exp(z_1^s - z_0^s)/exp(\frac{z_1^s - z_0^s}{\tau})$. The derivation clearly shows how PTLoss, through the appropriate selection of $\epsilon_{c,m}$, effectively subsumes temperature scaling. This capability to encompass temperature scaling further underlines the versatility of our approach.

Label Smoothing. We compare PTLoss with the label smoothing method and claim that label smoothing proposed in [29] is a special case of PTLoss. According to the implementation in Szegedy et al. [29], we can smooth the teacher labels in KD by

$$\mathbf{p}_c^{t_{ls}} = (1 - \delta)\mathbf{p}_c^t + \delta/2, \quad (14)$$

with a smoothing parameter δ. Starting from Eq. 5, we can replace the term $\mathbf{p}_c^t$ by its smooth version $\mathbf{p}_c^{t_{ls}}$. Then the original Eq. 5 with label smoothing becomes:

$$\ell_{KL}^{ls}\left(\mathbf{p}^t, \mathbf{p}^s\right) = -\mathbb{H}\left(\mathbf{p}^{t_{ls}}\right) + \sum_{c \in [C]} \mathbf{p}_c^{t_{ls}} \cdot (-\log \mathbf{p}_c^s) \quad (15)$$

For the entropy of the teacher output, the smooth version $\mathbb{H}\left(\mathbf{p}^{t_{ls}}\right)$ is different from the original $\mathbb{H}\left(\mathbf{p}^t\right)$ with only a constant C, which can be ignored when optimizing the loss function. Similarly, by letting $\ell_{KL}^{ls}\left(\mathbf{p}^t, \mathbf{p}^s\right) = \ell_{PT}\left(\mathbf{p}^t, \mathbf{p}^s\right)$, we obtain

$$\epsilon_{c,m} = \frac{\Delta \mathbf{p}_c^t}{m\mathbf{p}_c^t} \xrightarrow{\Delta \mathbf{p}_c^t = \delta/2 - \delta \mathbf{p}_c^t} \epsilon_{c,m} = \frac{\delta}{m}\left(\frac{1}{2\mathbf{p}_c^t} - 1\right). \quad (16)$$

In shows the connection between the two losses can be expressed through a specific $\epsilon_{c,m}$, which depends on the smoothing parameter δ. This derivation highlights that PTLoss generalizes the label

smoothing method and provides a more flexible framework that encompasses the effects of label smoothing.

In summary, we establish that PTLoss can effectively serve as a generalized framework that includes various techniques by manipulating the perturbation coefficients $\epsilon_{c,m}$. It is important to note that our goal is not to directly solve for the perturbation coefficients to make PTLoss equivalent to the alternative perturbation methods. Instead, we aim to show that our approach covers the loss shift space produced by them. To determine the real perturbation coefficient, we establish a principled searching method via the best proxy teacher, as demonstrated in the following section.

4 PROXY TEACHER AND THE PRINCIPLE OF SELECTING POLYNOMIAL COEFFICIENTS

In this section, we first present a theorem to show how the teacher model affects the gap of a student model's distillation empirical risk and its population risk (§ 4.1). Then, we demonstrate that using PTLoss implicitly transforms the original teacher model to a *proxy teacher* under the KL loss. Based on the above theorem, we know when this proxy teacher distribution is closer to the true distribution, we will have a better distilled student model (§ 4.2). Finally, we establish our principle of selecting the perturbation coefficients in PTLoss: searching the coefficients that lead to a proxy teacher closest to the empirical estimate of true distribution on a validation set (§ 4.3).

4.1 The Connection of the Teacher Model and the Risks of Student Model

Theorem 4.1. *Given a teacher model $\mathbf{p}^t$, an unlabeled distillation dataset $\mathcal{D}_u$ with an unknown true distribution $\mathbf{p}^*$, we have for any probability predictor $\mathbf{p}: \mathcal{X} \to \mathbb{R}^C$:*

$$\mathbb{E}\left[(\tilde{R}_{KL}(\mathbf{p};\mathbf{p}^t,\mathcal{D}_u) - R(\mathbf{p}))^2\right] \leq \frac{2}{N_u} \cdot \mathbb{V}\left[\mathbf{p}^t(x)^T \log(\mathbf{p}(x))\right] +$$
$$O\left((\mathbb{E}_x\left[\|\mathbf{p}^t(x) - \mathbf{p}^*(x)\|_2\right])^2 + \right.$$
$$\left. \mathbb{E}_x\left[\left(\mathbf{p}^t(x)^T \log \mathbf{p}^t(x)\right)^2\right]\right),$$

where $\mathbb{V}[\cdot]$ denotes the variance of a random variable.

We defer the detailed proofs of above theorem to Appendix 8.4 and focus on its implications here. We can see that the gap between a model $\mathbf{p}$'s distillation empirical risk and its population risk depends on three terms: (1) the variance of its KL distance to the teacher model $\mathbf{p}^t$, (2) the L_2 distance between the teacher model output distribution $\mathbf{p}^t$ and the true distribution $\mathbf{p}^*$, and (3) the entropy of the teacher distribution. In practice, obtaining a sizable unlabeled distillation set $\mathcal{D}_u$ is relatively straightforward, which leads to a large value of N_u. As a result, the first term (of order $O(1/N_u)$) will converge to 0 as N_u keeps increasing and the latter two terms (one quantifies the distance between teacher $\mathbf{p}^t$ and true $\mathbf{p}^*$, and the other quantifies the teacher's uncertainty) will dominate the risk gap. This observation also resonates with our intuition that an accurate and well-calibrated teacher yields better improved bounds on the generalization error of the student.

4.2 The Equivalence of Proxy Teacher under KL Loss and Original Teacher under PTLoss

The above theorem states that an ideal teacher model, when used in KL loss for distillation, should output a distribution as close to the true distribution as possible. In reality, however, the teacher model is usually fixed. Here, we show that using PTLoss for distillation can implicitly transform the original teacher to a *proxy teacher* under the KL loss. Namely, given the original teacher model $\mathbf{p}^t$ and a set of perturbation coefficients $\{\epsilon_{c,m}\}$ in PTLoss, we can obtain a proxy teacher $\mathbf{p}^{t_{px}}$ such that:

$$\tilde{R}_{KL}(\mathbf{p}^s; \mathbf{p}^{t_{px}}, \mathcal{D}_u) = \tilde{R}_{PT-M}(\mathbf{p}^s; \mathbf{p}^t, \mathcal{D}_u)$$
$$= \frac{1}{N_u} \sum_{n=1}^{N_u} \ell_{PT-M}(\mathbf{p}^t(x_n), \mathbf{p}^s(x_n)), \quad (17)$$

which establishes the equivalence of proxy teacher under KL loss and original teacher under PTLoss. With the proxy teacher $\mathbf{p}^{t_{px}}$, we aim to determine the best perturbation coefficients $\{\epsilon_{c,m}\}$. Note for each $\{\epsilon_{c,m}\}$, we can obtain a proxy teacher. We illustrate how we obtain the proxy teacher in the rest of this subsection, and discuss how to select the best perturbation coefficients in § 4.3.

Intuitively, the proxy teacher is derived by solving the below optimization problem:

$$\min_{\mathbf{p}^{t_{px}}} \quad \|\tilde{R}_{PT-M}(\mathbf{p}^s; \mathbf{p}^t, \mathcal{D}_u) - \tilde{R}_{KL}(\mathbf{p}^s; \mathbf{p}^{t_{px}}, \mathcal{D}_u)\|_2. \quad (18)$$

In practice, however, we do not need the above risk equivalence in Eq. 17 to hold for all possible student models $\mathbf{p}^s$. Instead, we focus on the minimizer of the left-hand side of Eq. 17 because it is practically close to the final learned student model. By substituting this minimizer $\mathbf{p}^s = \mathbf{p}^{t_{px}}$ into Eq. 18, the second term in the norm of Eq. 18 becomes 0, and the first term could be expanded by its definition in Eq. 17, we thus have the following objective:

$$\min_{\mathbf{p}^{t_{px}}} \left\| \frac{1}{N_u} \sum_{n=1}^{N_u} \left(\ell_{KL}(\mathbf{p}^t(x_n), \underline{\mathbf{p}^{t_{px}}(x_n)}) \right. \right. \quad (19)$$
$$\left. \left. + \sum_{c \in [C]} \mathbf{p}_c^t(x_n) \sum_{m=1}^{M} \epsilon_{c,m}(1 - \underline{\mathbf{p}_c^{t_{px}}(x_n)})^m \right) \right\|_2.$$

This objective enables us to solve $\mathbf{p}^{t_{px}}$ given $\mathbf{p}^t$ and $\{\epsilon_{c,m}\}$, where $\mathbf{p}^t$ is the teacher's output probability on the validation set, and $\{\epsilon_{c,m}\}$ is a given set of perturbation coefficients. However, this optimization problem is nonlinear and lacks a closed-form analytical solution. Consequently, we compute the $\mathbf{p}^{t_{px}}$ using the numerical approach[2]. Specifically, the optimization problem defined in Eq. 19 is solved via the algorithm 'scipy.optimize.fsolve', which is a hybrid method of the Newton-Raphson method and the Levenberg-Marquardt algorithm. For better numerical stability, we actually solve the equation in logit space (instead of the vanilla probability space) and use softmax function to map it back to the final probability. Another advantage of this approach is that we remove the probability constraint of $\mathbf{p}^{t_{px}}$. We also input the analytical form

---

[2]We use a hybrid algorithm of the Newton-Raphson method and the Levenberg-Marquardt algorithm as defined in 'scipy.optimize.fsolve' https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fsolve.html.

Algorithm 1 Automated Perturbation Coefficients Selection

Require: Validation set $\mathcal{D}_v$, Teacher model $\mathbf{p}^t$, Max perturbation order N_M, Max search trails N_k, Perturbation coefficient search space $\mathcal{S}$.

Initialize $\{\epsilon^*_{c,m}\} \leftarrow \{\}$, $\hat{Q}^* \leftarrow \infty$.

for $M = 1$ to N_M **do**

    for $k = 1$ to N_k **do**

        Randomly sample a perturbation coefficients set $\{\epsilon_{c,m}\} \in \mathcal{S}$.

        Solve the proxy teacher $\mathbf{p}^{t_{px}}$ given $\{\epsilon_{c,m}\}$ and $\mathbf{p}^t$ via Eq. 19.

        Compute the quality score of perturbation coefficients $\hat{Q}(\{\epsilon_{c,m}\})$ via Eq. 20.

        if $\hat{Q}(\{\epsilon_{c,m}\}) < \hat{Q}^*$ **then**

            $\hat{Q}^* \leftarrow \hat{Q}(\{\epsilon_{c,m}\})$, $\{\epsilon^*_{c,m}\} = \{\epsilon_{c,m}\}$.

        end if

    end for

end for

return $\{\epsilon^*_{c,m}\}$.

---

of the Jacobian of our optimization objective into the solver (via the 'fprime' parameter) and set the initial estimate of $\mathbf{p}^{t_{px}}$ to be the original teacher $\mathbf{p}^t$ (via the 'x0' parameter). For all the other parameters in 'scipy.optimize.fsolve', we use their default values.

We have also considered an alternative solution to this optimization problem, which involves defining a parameterized function $g_\theta(\cdot): [0,1]^C \rightarrow [0,1]^C$ that explicitly transforms the original teacher to the proxy teacher, namely $g_\theta(\mathbf{p}^t) = \mathbf{p}^{t_{px}}$. We would then find the best θ minimizing the above objective (possibly via gradient-based methods). This approach leads to a smooth proxy teacher but also introduces bias from the function class defined by θ. Therefore, we leave it to future work and resort to the numerical approach in this study.

4.3 Selecting Perturbation Coefficients via the Best Proxy Teacher

For each candidate set of perturbation coefficients $\{\epsilon_{c,m}\}$ in PTLoss, we can find a corresponding proxy teacher and compute its risk deviation upper bound according to theorem 4.1. In practice, the size of distillation set N_u is typically large and thus we can omit the $O(1/N_u)$ variance term. Furthermore, since the ground truth distribution $\mathbf{p}^*$ is unknown, we use an unbiased estimator to replace it. Finally, we replace the expectation by the sample mean and define the empirical risk below:

$$
\hat{Q}(\{\epsilon_{c,m}\}) = \left(\frac{1}{N_v} \sum_{n=1}^{N_v} \left[\|\mathbf{p}^{t_{px}}(x_n) - \mathbf{y}_n\|_2 \right] \right)^2 \\
+ \frac{1}{N_v} \sum_{n=1}^{N_v} \left[\left(\mathbf{p}^{t_{px}}(x_n)^T \log \mathbf{p}^{t_{px}}(x_n) \right)^2 \right],
\tag{20}
$$

where N_v is the size of validation set and $\mathbf{y}_n$ is a one-hot label vector of x_n, serving as the unbiased estimation of $\mathbf{p}^*(x_n)$. We use $\hat{Q}(\{\epsilon_{c,m}\})$ as a "quality score" for each candidate coefficients set. Users can define a search space of $\{\epsilon_{c,m}\}$ and we will pick the optimal $\{\epsilon^*_{c,m}\}$ that minimizes $\hat{Q}$. We present the pseudo-code for selecting perturbation coefficients in Algorithm 1, and the search time for perturbation coefficients is detailed in Appendix 8.5.
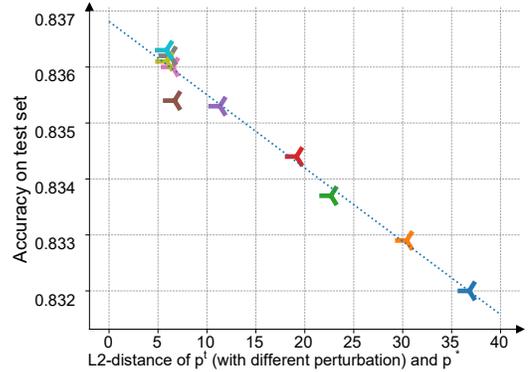
5 EXPERIMENTS

In this section, we first conduct experiments on a synthetic dataset to verify our assumption that the teacher outputting a distribution closer to the ground truth distribution leads to a better student (§5.1). Then, we present our main results on 6 real-world NLP datasets (§5.2). Moreover, we show how the proxy teacher enhances the distillation process(§5.3). In the appendix, we further evaluate the performance of PTLoss on CIFAR-100 to show its potential in computer vision tasks (§5.4).

5.1 Experiments on Synthetic Gaussian Dataset
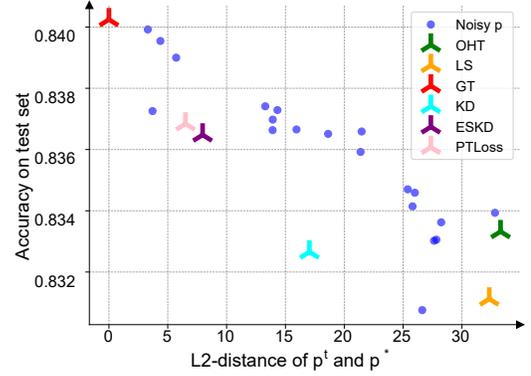
(a) OHT - one-hot training; LS - label smoothing; GT - ground truth; KD - knowledge distillation; ESKD: early-stopped KD [27]; PTLoss: here we use 3-order perturbation.

(b) Correlation between the L_2-distance (between the teacher model $\mathbf{p}^t$ with different levels of perturbations and the ground truth $\mathbf{p}^*$) and the test accuracy of the student model.

Figure 3: Experiments on a synthetic Gaussian dataset.

We first conduct an illustrative experiment with a synthetic dataset where the ground truth distribution $\mathbf{p}^*(x)$ is known. Specifically, we follow [27] to generate 10^5 examples from a mixture of Gaussian distribution and train an MLP with 3 hidden layers on this synthetic dataset. This is a 3-class toy Gaussian dataset with

10k data points, divided into training, validation, and test sets with a split ratio $[0.9, 0.05, 0.05]$. The underlying model in this set of experiments is a 2-layer MLP with ReLU activation, and the hidden size is 128 for each layer. We set the learning rate as 5×10^{-4}, the batch size as 32, and the training epochs as 100.

The sampling process is implemented as follows: We first choose the label y using a uniform distribution across all the 3 classes. Next, we sample $x|_{y=k} \sim \mathcal{N}\left(\mu_k, \sigma^2 I\right)$ as the input signal. Here $\sigma = 2$ and μ_k is a 30-dim vector with entries randomly selected from $\{-1, 0, 1\}$.

We compare PTLoss with 4 baselines: one-hot supervision (OHT), label smoothing (LS), standard knowledge distillation (KD), and early-stopped knowledge distillation (ESKD) (see details in below § 5.2). As illustrated in Fig. 3a, the quality of the distilled student improves as the L_2-distance between the teacher distribution and the ground truth distribution decreases. On this synthetic Gaussian dataset, PTLoss also outperforms the baselines after adding a 3-order perturbation.

In Fig. 3b, we sample 10 proxy teachers in different stages of the perturbation coefficient searching process (§4.3) and compare their results. It is clear that a teacher model with a smaller L_2-distance to the ground truth distribution can lead to a better student model. This observation verifies our hypothesis in Eq. 20 — searching a proxy teacher closer to the ground truth distribution can reduce the empirical deviation and improve the distilled student model.

5.2 Experiments on Natural Language Datasets

Tasks and Datasets. We conduct our main experiments on 6 natural language datasets, including:
(1) CoLA [32] for linguistic acceptability;
(2) MNLI [35] for multi-genre natural language inference;
(3) MRPC [8] for paraphrase similarity matching;
(4) RTE [31] for textual entailment inference;
(5) SST-2 [31] for sentiment analysis;
(6) BoolQ [6] for boolean question answering.
We list the detailed dataset statistics in the Appendix 8.1.

Model Architectures. For the teacher model, we choose the T5 architecture [26] and select two teacher models of different scales. Specifically, we use T5-xxl with 11 billion parameters and T5-large with 770 million parameters. For the student model, we use the BERT-base model [7] with 110 million parameters.

Compared Methods. We compare our proposed PTLoss with the following baselines:
(1) Standard KL loss [17]: adopts standard KL divergence loss for knowledge distillation;
(2) Temperature scaling [11]: scales the teacher output logits via a temperature hyper-parameter;
(3) Label smoothing [29]: smooths the teacher output class probabilities by a small scalar;
(4) Focal loss [19]: modulates the cross-entropy loss to focus on hard examples;
(5) Flooding [13]: a regularization method to intentionally prevent further reduction of the training loss;
(6) CRD [30]: uses a contrastive objective in knowledge distillation;
(7) AnnealingKD [14]: feeds the rich information provided by the teacher's soft-targets incrementally;

(8) FilterKD [27]: trains the student from the smoothed predictions of the teacher network;
(9) MetaDistill [40]: evolves the teacher network with the feedback from the distilled student in a meta learning framework.

For all the baselines, we conduct an exhaustive hyper-parameter search on the validation set. For our own PTLoss method, we set its perturbation order $M = 5$ and use the proxy teacher-based method to search its perturbation coefficients (§4.3). See Appendix 8.2 for more details. We run each method with three different random seeds and report its average performance.

Main Results. Table 1 shows the main results on 6 NLP datasets. We have the following observations: (1) PTLoss outperforms on 11 out of the total 12 tasks, achieving the best average performance across the board. The only exception is MetaDistill, which tops the results on the MRPC when using the T5-xxl teacher and ties with PTLoss on MNLI when using the T5-large teacher. (2) The advantages offered by PTLoss are robust, regardless of the scale of the teacher model. Notably, as the disparity in scale between the teacher model and student model reduces, the performance gap between them also narrows. (3) In comparison to vanilla KD, which utilizes the standard KL, PTLoss showcases significant enhancement. Specifically, it exceeds standard KL by an average of 2.8% and 2.9%, respectively. (4) Surveying the baseline methods, MetaDistill stands out, securing the second-highest performance across most tasks. On the whole, the cluster of KD methods generally outstrips the simple regularization methods.

5.3 Proxy Teacher Analysis

Correlation between teacher's distance to ground truth and student's performance. To explore where PTLoss's performance gains come from, we train multiple teacher models on the BoolQ dataset and distill them into the student models. Fig. 4a shows the student model performance on the test set is highly correlated with the distance between the teacher model's output distribution and the ground truth distribution on the validation set. This result resonates with our findings from synthetic datasets(§ 5.1), confirming that, on real-world datasets, a teacher model with a predictive distribution closer to the ground truth can produce a more effectively distilled student.

Effectiveness of Perturbation Coefficients Search. We continue to validate the effectiveness of the proxy teacher-based perturbation coefficients selection method using MNLI as a representative dataset. Specifically, we vary the perturbation order M from 1 to 5 and report the performance of the student models distilled via PTLoss with different perturbation coefficients. These coefficients are obtained either by minimizing the empirical risk deviation of proxy teacher (c.f. Eq. 19) or via random sampling from the space of $[-1, 10]^M$. As shown in Fig. 4b, the coefficients obtained from our proxy teacher based method can achieve consistent improvements over the random coefficients. If we just randomly set the perturbation coefficients, the student performance can drop by up to 1.2%. Also, by comparing different perturbation orders, we find that the higher the perturbation order, the greater the performance differences. This is because in the higher-dimension space, it is harder for random search to get a set of appropriate perturbation coefficients,

Table 1: Main results on natural language datasets. The student model (BERT-base) is distilled from teacher models of different sizes (T5-xxl and T5-large). All results are averaged over three runs. The bolded numbers indicate the best results, while the underscore "_" denotes the second-best results.

Method	CoLA (Matt.)	MNLI (Acc.)	MRPC (F1)	RTE (Acc.)	SST-2 (Acc.)	BoolQ (Acc.)	Average
Teacher T5-xxl	71.5	94.7	92.4	92.2	96.4	89.1	89.4
Standard KL	58.8 ± 0.3	90.3 ± 0.2	88.3 ± 0.4	78.1 ± 0.1	89.2 ± 0.3	69.5 ± 0.2	79.0
Temp. Scaling	59.4 ± 0.3	90.7 ± 0.1	88.9 ± 0.3	79.6 ± 0.3	89.4 ± 0.2	<u>72.0</u> ± 0.3	80.0
Label Smoothing	59.3 ± 0.6	90.6 ± 0.4	89.2 ± 0.7	79.2 ± 0.4	89.9 ± 0.3	68.6 ± 0.4	79.6
Focal	59.2 ± 0.4	90.7 ± 0.3	88.7 ± 0.9	80.4 ± 0.4	89.3 ± 0.3	68.2 ± 1.5	79.6
Flooding	58.9 ± 0.5	90.6 ± 0.4	89.6 ± 0.6	80.2 ± 0.7	89.3 ± 0.4	69.3 ± 0.5	79.7
CRD	59.5 ± 0.5	90.5 ± 0.3	90.6 ± 0.4	81.1 ± 0.2	<u>89.6</u> ± 0.3	71.8 ± 0.6	80.5
Annealing KD	59.8 ± 0.3	90.7 ± 0.3	90.0 ± 0.5	80.7 ± 0.2	89.3 ± 0.5	70.7 ± 0.5	80.2
FilterKD	59.2 ± 0.4	90.7 ± 0.2	89.5 ± 0.3	80.4 ± 0.3	89.3 ± 0.2	69.6 ± 0.9	79.8
MetaDistill	<u>60.4</u> ± 0.2	<u>90.8</u> ± 0.3	**91.4** ± 0.4	<u>81.3</u> ± 0.1	89.5 ± 0.2	71.9 ± 0.7	<u>80.9</u>
PTLoss(ours)	**61.2** ± 0.3	**91.1** ± 0.1	<u>91.2</u> ± 0.3	**83.5** ± 0.2	**90.3** ± 0.1	**73.1** ± 0.5	**81.8**
Teacher T5-large	61.4	93.6	92.1	87.2	95.5	77.9	84.6
Standard KL	54.8 ± 0.2	90.0 ± 0.1	87.8 ± 0.3	77.6 ± 0.2	88.8 ± 0.1	69.5 ± 0.2	78.1
Temp. Scaling	55.6 ± 0.3	90.4 ± 0.1	88.7 ± 0.2	79.4 ± 0.2	89.2 ± 0.5	70.4 ± 0.6	79.1
Label Smoothing	56.4 ± 0.4	90.6 ± 0.2	89.2 ± 0.6	79.2 ± 0.4	89.2 ± 0.4	69.1 ± 1.2	79.1
Focal	56.0 ± 0.2	90.3 ± 0.1	88.4 ± 0.5	79.9 ± 0.4	89.3 ± 0.5	68.9 ± 0.5	78.8
Flooding	57.8 ± 0.3	90.0 ± 0.6	89.5 ± 0.4	79.5 ± 0.4	89.0 ± 0.4	68.9 ± 0.6	79.3
CRD	58.2 ± 0.3	90.2 ± 0.4	<u>89.8</u> ± 0.3	80.3 ± 0.2	<u>89.4</u> ± 0.4	<u>70.5</u> ± 0.5	79.7
Annealing KD	58.3 ± 0.2	90.4 ± 0.3	<u>89.8</u> ± 0.5	79.9 ± 0.1	<u>89.4</u> ± 0.4	69.7 ± 0.4	79.6
FilterKD	56.7 ± 0.3	90.2 ± 0.2	89.1 ± 0.4	78.8 ± 0.2	89.2 ± 0.3	69.2 ± 0.6	78.9
MetaDistill	<u>58.6</u> ± 0.2	**90.7** ± 0.3	89.6 ± 0.1	<u>81.0</u> ± 0.1	89.3 ± 0.2	70.4 ± 0.2	<u>80.1</u>
PTLoss(ours)	**60.5** ± 0.2	**90.7** ± 0.1	**91.1** ± 0.4	**82.7** ± 0.1	**90.0** ± 0.2	**71.0** ± 0.3	**81.0**
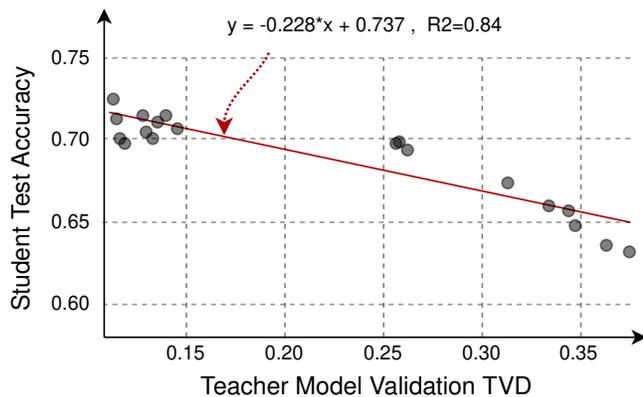
(a) Correlation between the teacher-to-ground-truth distance and the student model performance. Experiments are conducted on BoolQ.
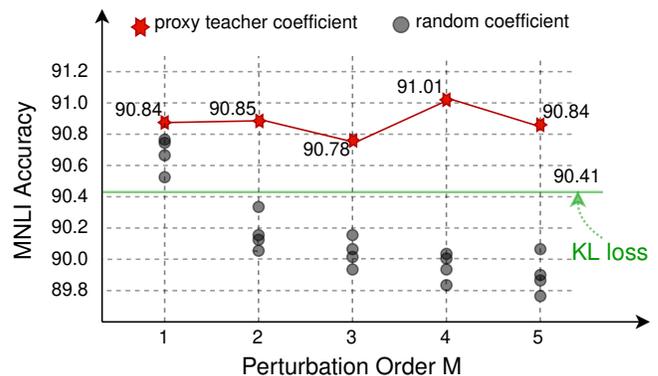
(b) The Proxy Teacher method v.s. random search for the perturbation coefficient selection. Experiments are conducted on MNLI.

Figure 4: PTLoss analysis.

Table 2: Results on CIFAR-100 dataset.

	MobileNetV2	ShuffleNetV2	ResNet18	GoogleNet	DenseNet121	ResNeXt29
Baseline	68.38	70.34	75.87	78.72	79.04	81.03
Tf-KD	70.14 ± 0.08	71.64± 0.17	76.60± 0.06	79.62± 0.43	79.54±0.16	80.75± 0.13
PTLoss	**70.62 ± 0.16**	**71.97± 0.10**	**77.55 ± 0.06**	**80.22 ± 0.11**	**80.22± 0.11**	**81.83 ± 0.21**

which makes the random PTLoss even worse than the standard KL loss. Conversely, equipped with the perturbation coefficients obtained via proxy teacher, PTLoss can significantly outperform the underlying KL loss.

5.4 Experiments on the CIFAR-100 dataset

To evaluate PTLoss on diverse tasks, we conduct this set of experiments on the CIFAR-100 dataset. We follow [36] to get the baseline results on the studied networks. Then we re-implement Tf-KD in [36] as we don't have access to the ground truth data during the distillation stage. Specifically, The Tf-KD implementation is modified from Tf-KD_self to be incorporated into our setting: we modified Eq.(7) in [36] as $L_{self} = D_{KL}(p^t_\tau, p_\tau)$, where p_τ and p^t_τ are the output probability of the student model and the pre-trained student model, reshaped by a temperature τ. For our methods, we simply add 1-order perturbation to p^t_τ, the ϵ is selected from $\{0.1, 0.2, 0.5\}$. From Table 2, we observe that PTLoss can still outperform those baselines, which shows the applicability of PTLoss on computer vision tasks.

6 RELATED WORK

6.1 Knowledge Distillation

Knowledge distillation was first proposed in [4] to compress the large models to smaller, faster models without a significant performance drop. Hinton et al. [11] generalized this technique by introducing a temperature parameter to smooth the teacher model prediction. Tian et al. [30] employed contrastive learning to train the student model. Later, Yuan et al. [36] explored the connection between KD and label smoothing, Jafari et al. [14] and Chen et al. [5] studied the feeding mechanism of the teacher's knowledge. Zhao et al. [38] decoupled the classical loss to target classes and non-target classes for KD efficiency and flexibility. Ren et al. [27] investigated supervisory signals and proposed to average teacher outputs for KD stability, while Zhou et al. [40] evolves the teacher model with the student feedback in a meta learning framework. Additionally, there is also significant research on applying KD to text generation tasks, as demonstrated by [1, 10, 16, 33, 37]. However, the exploration of KD in text generation represents a distinct and extensive area of study, it is beyond the scope of this paper because our focus is on KD within language understanding tasks.

6.2 Distillation Theory

Concurrent with the empirical success of knowledge distillation, numerous works aim to understand its mechanisms. Hinton et al. [11] suggest that teacher's soft labels offer "dark knowledge" through weights on incorrect labels. Menon et al. [22] present a statistical view, observing that a good teacher model should be Bayesian to

reduce the student objective variance. Stanton et al. [28] highlight discrepancies between teacher and student output distributions and emphasize the optimization challenge in distillation. While more recent studies [2, 12, 15, 39] explore distillation from several various angles, a gap remains between the theoretical analysis and the improved distillation techniques.

6.3 Loss Function Design

Our work also relates to loss function design. Lin et al. [19] propose a modification of the cross-entropy loss function, reshaping it to focus more on hard examples and address issues of data imbalance. Leng et al. [18] expand cross-entropy loss and focal loss into a linear combination of polynomial functions, primarily studying Poly-1 formulation on computer vision tasks while avoiding issues with high-order polynomial hyper-parameter searches. TaylorGLO [9] utilizes Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to optimize multivariate Taylor parameterization of a loss function and learning rate schedule. However, it lacks a principled analysis of the performance gains following perturbations. In contrast, our work provides both theoretical and empirical evidence supporting the necessity of incorporating perturbations into the Knowledge Distillation (KD) learning objective, especially when employing a high-fidelity teacher for effective student supervision.

7 CONCLUSIONS AND FUTURE WORK

In this study, we proposed a novel knowledge distillation loss PTLoss which implicitly shifts the teacher model output distribution to a high-fidelity one for student model training. We also established connections between PTLoss and other loss functions by demonstrating that PTLoss can subsume the others while providing more flexible adjustments to teacher models. We theoretically showed how the teacher model affects the student model risks and presented a principled method to systematically search perturbation coefficients. Extensive experiments on multiple tasks verified our proposed theory and validated the effectiveness of PTLoss.

While PTLoss enables better KD by creating a proxy teacher closer to the ground truth distribution, we focus on the single-teacher-single-student setting in this work. It is worth exploring how this approach can be extended to ensemble KD involving multiple teachers or students. Additionally, although the proposed coefficients selection method provides a principal way to determine the perturbation hyperparameters, it remains challenging to scale up the number of classes and the perturbation order. Future work could benefit from developing scalable methods for hyperparameter search, enabling rapid determination of perturbation coefficients even in high-dimensional spaces with numerous classes or high perturbation orders.

REFERENCES

[1] Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. GKD: Generalized Knowledge Distillation for Auto-regressive Sequence Models. *arXiv preprint arXiv:2306.13649* (2023).

[2] Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. In *ICLR*.

[3] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. 2005. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics* 9 (2005), 323–375.

[4] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 535–541.

[5] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2021. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5008–5017.

[6] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044* (2019).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. https://aclanthology.org/I05-5002

[9] Santiago Gonzalez and Risto Miikkulainen. 2021. Optimizing loss functions through multi-variate taylor polynomial parameterization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 305–313.

[10] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge Distillation of Large Language Models. *arXiv preprint arXiv:2306.08543* (2023).

[11] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[12] Daniel Hsu, Ziwei Ji, Matus Telgarsky, and Lan Wang. 2021. Generalization bounds via distillation. *arXiv preprint arXiv:2104.05641* (2021).

[13] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? *arXiv preprint arXiv:2002.08709* (2020).

[14] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. 2021. Annealing Knowledge Distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2493–2504.

[15] Guangda Ji and Zhanxing Zhu. 2020. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. *Advances in Neural Information Processing Systems* 33 (2020), 20823–20833.

[16] Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947* (2016).

[17] Solomon Kullback. 1959. Statistics and information theory.

[18] Zhaoqi Leng, Mingxing Tan, Chenxi Liu, Ekin Dogus Cubuk, Jay Shi, Shuyang Cheng, and Dragomir Anguelov. 2022. PolyLoss: A Polynomial Expansion Perspective of Classification Loss Functions. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gSdSJoenupI

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[20] Boxiao Liu, Shenghan Zhang, Guanglu Song, Haihang You, and Yu Liu. 2021. Rectifying the Data Bias in Knowledge Distillation. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 1477–1486. https://doi.org/10.1109/ICCVW54120.2021.00171

[21] Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. 2021. Teacher's pet: understanding and mitigating biases in distillation. *arXiv preprint arXiv:2106.10494* (2021).

[22] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. 2021. A statistical perspective on distillation. In *International Conference on Machine Learning*. PMLR, 7632–7642.

[23] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When Does Label Smoothing Help?. In *NeurIPS*.

[24] OpenAI. 2022. ChatGPT.

[25] OpenAI. 2023. GPT-4 Technical Report.

[26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.

[27] Yi Ren, Shangmin Guo, and Danica J Sutherland. 2022. Better Supervisory Signals by Observing Learning Paths. *arXiv preprint arXiv:2203.02485* (2022).

[28] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. 2021. Does knowledge distillation really work? *Advances in Neural Information Processing Systems* 34 (2021), 6906–6919.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[30] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699* (2019).

[31] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[32] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural Network Acceptability Judgments. *arXiv preprint arXiv:1805.12471* (2018).

[33] Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-Divergence Minimization for Sequence-Level Knowledge Distillation. *arXiv preprint arXiv:2307.15190* (2023).

[34] Jennifer C. White and Ryan Cotterell. 2021. Examining the Inductive Bias of Neural Language Models with Artificial Languages. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:235293810

[35] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[36] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3903–3911.

[37] Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Haorui Wang, Zhen Qin, Feng Han, Jialu Liu, Simon Baumgartner, Michael Bendersky, and Chao Zhang. 2024. PLaD: Preference-based Large Language Model Distillation with Pseudo-Preference Pairs. *arXiv preprint arXiv:2406.02886* (2024).

[38] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 11953–11962.

[39] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. 2021. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv preprint arXiv:2102.00650* (2021).

[40] Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022. BERT Learns to Teach: Knowledge Distillation with Meta Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7037–7049.

Table 3: Dataset Statistics

Dataset	Task	Train	Distillation	Dev	Test
CoLA	Linguistic Acceptability	8.5k	8.5k	3k	1k
MNLI	Natural Language Inference	58.9k	314k	19.6k	9.8k
MRPC	Paraphrase Similarity Matching	3.7k	3.7k	1k	1.7k
RTE	Textual Entailment Inference	2.5k	2.5k	0.8k	3k
SST-2	Sentiment Analysis	6.7k	53.8k	6.7k	872
BoolQ	Boolean Question Answering	2.5k	5.9k	1k	3.2k

Table 4: The search range of hyper-parameters.

Hyper-parameter	Search Range
Learning Rate	$\{2, 3, 5\} \times 10^{-5}$
Batch Size	$\{8, 16, 32, 64, 128, 256\}$
Temperature T	$\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10\}$
Label Smoothing δ	$\{0.02, 0.05, 0.1, 0.15, 0.2\}$
Focal Loss τ	$\{0.1, 0.2, 0.5, 1, 2.0, 5.0\}$
Random PTLoss $\epsilon_{c,m}$	$[-1, 10]$

8 APPENDIX

8.1 Dataset Statistics

We list the dataset statistics in Table 3.

8.2 Hyper-parameters

We list the search range of hyperparamters in Table 4. The search for batch size and learning rate is applied to all the methods. And for each baseline, we search for the best baseline-specific hyper-parameters.

8.3 Connections between PTLoss and other Perturbation Methods

In Section 4.1, we have presented the connection between PTLoss and other perturbation methods. The connection between PTLoss and the standard KL loss is direct, and it has been clearly presented how PTLoss subsumes the focal loss. Now we detail the derivation of the connection between PTLoss and temperature scaling and label smoothing here.

Temperature Scaling. We compare PTLoss with temperature scaling and claim that PTLoss subsumes it with appropriate approximation. As described in Hinton et al. [11], the logits are adjusted by a temperature to control sharpness or smoothness of the probability distribution.

$$\mathbf{p}_{0,\tau} = \frac{exp(z_0/\tau)}{exp(z_0/\tau) + exp(z_1/\tau)}, \quad (21)$$

where τ is the temperature and z_c is the logits. Here we use binary classification (*i.e.,* $c = 0, 1$) for the derivation simplicity without loss of generality. Denote the probability without temperature scaling as $\mathbf{p}_0$, we have

$$\frac{\mathbf{p}_{0,\tau}}{\mathbf{p}_0} = \frac{exp(z_0/\tau)}{exp(z_0/\tau) + exp(z_1/\tau)} \times \frac{exp(z_0) + exp(z_1)}{exp(z_0)}$$
$$= \frac{1 + exp(z_1 - z_0)}{1 + exp(\frac{z_1 - z_0}{\tau})}. \quad (22)$$

In practice, we have

$$\frac{1 + exp(z_1 - z_0)}{1 + exp(\frac{z_1 - z_0}{\tau})} \approx 1 \quad \text{or} \quad \frac{exp(z_1 - z_0)}{exp(\frac{z_1 - z_0}{\tau})} \quad (23)$$

because $|z_1 - z_0| \gg 0$. Then we have

$$\mathbf{p}_{0,\tau} \approx \mathbf{p}_0 \quad \text{or} \quad \frac{exp(z_1 - z_0)}{exp(\frac{z_1 - z_0}{\tau})} \mathbf{p}_0. \quad (24)$$

For the first case where $\mathbf{p}_{0,\tau} \approx \mathbf{p}_0$, we omit the discussion as it aligns with the standard KL loss. For the second case, we proceed to draw its connection with PTLoss as follows. We denote $\mathbf{p}_{\tau,c}^t$, $\mathbf{p}_{\tau,c}^s$ as the teacher, student probability scaled by temperature τ. Incorporating temperature scaling, the KL loss can be formulated as:

$$\ell_{KL}^{temp}\left(\mathbf{p}_\tau^t, \mathbf{p}_\tau^s\right) = -\mathbb{H}\left(\mathbf{p}_\tau^t\right) + \sum_{c\in[C]} \mathbf{p}_{\tau,c}^t \cdot (-\log \mathbf{p}_{\tau,c}^s), \quad (25)$$

Substituting $\mathbf{p}_{\tau,c}^t$ and $\mathbf{p}_{\tau,c}^s$ in Eq. 25 using Eq. 24, we obtain

$$\ell_{KL}^{temp}\left(\mathbf{p}_\tau^t, \mathbf{p}_\tau^s\right) = -\mathbb{H}\left(\mathbf{p}_\tau^t\right) + \sum_{c\in[C]} a \cdot \mathbf{p}_c^t \cdot [-\log(b \cdot \mathbf{p}_c^s)], \quad (26)$$

where $a = exp(z_1^t - z_0^t)/exp(\frac{z_1^t - z_0^t}{\tau})$, and $b = exp(z_1^s - z_0^s)/exp(\frac{z_1^s - z_0^s}{\tau})$.

Comparing above Eq. 26 with Eq. 5, we can set

$$\sum_{m=1}^\infty (\frac{1}{m} + \epsilon_{c,m}) \cdot (1 - \mathbf{p}_c^s)^m = a[\sum_{m=1}^\infty \frac{1}{m} \cdot (1 - \mathbf{p}_c^s)^m + \sum_{m=1}^\infty \frac{1}{m} \cdot (1 - b)^m]. \quad (27)$$

It leads to

$$\epsilon_{c,m} = \frac{a}{m} \cdot (1 + (\frac{1 - b}{1 - \mathbf{p}_c^s})^m) - \frac{1}{m}, \quad (28)$$

which indicates PTLoss can encompass the temperature-scaled distillation loss by setting a group of appropriate perturbation coefficients.

Label Smoothing. We compare PTLoss with the label smoothing method and claim that label smoothing proposed in [29] is a special case of PTLoss. According to the implementation in Szegedy et al. [29], we can smooth the teacher labels in KD by

$$\mathbf{p}_c^{t_{ls}} = (1 - \delta)\mathbf{p}_c^t + \delta/2, \tag{29}$$

with a smoothing parameter δ. Starting from Eq. 5, we can replace the term $\mathbf{p}_c^t$ by its smooth version $\mathbf{p}_c^{t_{ls}}$. Then the original Eq. 5 with label smoothing becomes:

$$\ell_{KL}^{ls}\left(\mathbf{p}^t, \mathbf{p}^s\right) = -\mathbb{H}\left(\mathbf{p}^{t_{ls}}\right) + \sum_{c \in [C]} \mathbf{p}_c^{t_{ls}} \cdot (-\log \mathbf{p}_c^s) \tag{30}$$

For the entropy of the teacher output, the smooth version $\mathbb{H}\left(\mathbf{p}^{t_{ls}}\right)$ is different from the original $\mathbb{H}\left(\mathbf{p}^t\right)$ with only a constant C, which can be ignored when optimizing the loss function. We introduce $\Delta \mathbf{p}_c^t = \delta/2 - \delta \mathbf{p}_c^t$ and replace all the $\mathbf{p}_c^{t_{ls}}$ in Eq. 30 by $\mathbf{p}_c^{t_{ls}} = \mathbf{p}_c^t + \Delta \mathbf{p}_c^t$, then we get:

$$\ell_{KL}^{ls}\left(\mathbf{p}^t, \mathbf{p}^s\right) = -\mathbb{H}\left(\mathbf{p}^{t_{ls}}\right) + \sum_{c \in [C]} (\mathbf{p}_c^t + \Delta \mathbf{p}_c^t) \cdot (-\log \mathbf{p}_c^s) \tag{31}$$

Similarly, we let $\ell_{KL}^{ls}\left(\mathbf{p}^t, \mathbf{p}^s\right) = \ell_{PT}\left(\mathbf{p}^t, \mathbf{p}^s\right)$, it yields

$$\sum_{c \in [C]} (\mathbf{p}_c^t + \Delta \mathbf{p}_c^t) \sum_{m=1}^{\infty} \frac{1}{m} \cdot (1 - \mathbf{p}_c^s)^m = \sum_{c \in [C]} \mathbf{p}_c^t \sum_{m=1}^{\infty} (\frac{1}{m} + \epsilon_{c,m}) \cdot (1 - \mathbf{p}_c^s)^m. \tag{32}$$

We obtain

$$\epsilon_{c,m} = \frac{\Delta \mathbf{p}_c^t}{m \mathbf{p}_c^t} \xrightarrow{\Delta \mathbf{p}_c^t = \delta/2 - \delta \mathbf{p}_c^t} \epsilon_{c,m} = \frac{\delta}{m}(\frac{1}{2\mathbf{p}_c^t} - 1). \tag{33}$$

In summary, the connection between the two losses can be expressed through a specific $\epsilon_{c,m}$, which depends on the smoothing parameter δ. This derivation highlights that PTLoss generalizes the label smoothing method and provides a more flexible framework that encompasses the effects of label smoothing.

8.4 Proof of Theorem 1

The theorem 1 states that given a teacher model $\mathbf{p}^t$, an unlabeled distillation dataset $\mathcal{D}_u$ with an unknown true distribution $\mathbf{p}^*$, we have for any probability predictor $\mathbf{p} : \mathcal{X} \to \mathbb{R}^C$:

$$\mathbb{E}\left[(\tilde{R}_{KL}(\mathbf{p}; \mathbf{p}^t, \mathcal{D}_u) - R(\mathbf{p}))^2\right] \leq \frac{2}{N_u} \cdot \mathbb{V}\left[\mathbf{p}^t(x)^T \log(\mathbf{p}(x))\right] +$$
$$O\left(\left(\mathbb{E}_x\left[\|\mathbf{p}^t(x) - \mathbf{p}^*(x)\|_2\right]\right)^2 +$$
$$\mathbb{E}_x\left[\left(\mathbf{p}^t(x)^T \log \mathbf{p}^t(x)\right)^2\right]\right),$$

where $\mathbb{V}[\cdot]$ denotes the variance of a random variable.

Proof. We first rewrite the population risk $R(\mathbf{p})$ with cross-entropy loss l_{CE} plugged in as follow:

$$R(\mathbf{p}) = \mathbb{E}_{(x,y)}\left[\ell(y, \mathbf{p}(x))\right]$$
$$= \mathbb{E}_x\left[\mathbb{E}_{y|x}\left[\ell(y, \mathbf{p}(x))\right]\right] \tag{34}$$
$$= \mathbb{E}_x\left[\mathbf{p}^*(x)^T(-\log(\mathbf{p}(x)))\right].$$

Then, we write out the distillation empirical distillation risk defined in Eq. 3 and have:

$$\tilde{R}_{KL}(\mathbf{p}; \mathbf{p}^t, \mathcal{D}_u) - R(\mathbf{p}) = \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbf{p}^t(x_n)^T(-\log(\mathbf{p}(x_n)))$$
$$+ \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbf{p}^t(x_n)^T \log(\mathbf{p}^t(x_n)) - \mathbb{E}_x\left[\mathbf{p}^*(x)^T(-\log(\mathbf{p}(x)))\right]. \tag{35}$$

We let

$$\Delta \doteq \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbf{p}^t(x_n)^T(-\log(\mathbf{p}(x_n))) - \mathbb{E}_x\left[\mathbf{p}^*(x)^T(-\log(\mathbf{p}(x)))\right],$$

and

$$H \doteq \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbf{p}^t(x_n)^T \log(\mathbf{p}^t(x_n)),$$

then

$$\mathbb{E}\left[(\tilde{R}_{KL}(\mathbf{p}; \mathbf{p}^t, \mathcal{D}_u) - R(\mathbf{p}))^2\right] = \mathbb{E}\left[(\Delta + H)^2\right]$$
$$\leq 2\mathbb{E}\left[\Delta^2\right] + 2\mathbb{E}\left[H^2\right]$$
$$= 2\mathbb{V}\left[\Delta\right] + 2\mathbb{E}\left[\Delta\right]^2 + 2\mathbb{E}\left[H^2\right] \tag{36}$$

where the second line is by the inequality $(a + b)^2 \leq 2a^2 + 2b^2$ and the linearity of expectation, and the third line is by $\mathbb{E}\left[\Delta^2\right] = \mathbb{V}\left[\Delta\right] + \mathbb{E}\left[\Delta\right]^2$. Observe that

$$\mathbb{E}\left[\Delta\right] = \mathbb{E}_x\left[(\mathbf{p}^t(x) - \mathbf{p}^*(x))^T(-\log(\mathbf{p}^t(x_n)))\right]$$
$$\leq \mathbb{E}_x\left[\|\mathbf{p}^t(x) - \mathbf{p}^*(x)\|_2 \cdot \|\log(\mathbf{p}^t(x_n))\|_2\right]$$
$$\leq \mathbb{E}_x\left[\|\mathbf{p}^t(x) - \mathbf{p}^*(x)\|_2 \cdot c_1 \cdot \|\log(\mathbf{p}^t(x_n))\|_\infty\right] \tag{37}$$
$$\leq c_2 \mathbb{E}_x\left[\|\mathbf{p}^t(x) - \mathbf{p}^*(x)\|_2\right],$$

where the second line is by the Cauchy-Schwartz inequality, the third line is by the equivalence of norms with a constant c_1, and the last line is by the boundedness of the log loss term[3].

Furthermore, we notice the $R(\mathbf{p})$ term in the above Δ is a constant and thus have:

$$\mathbb{V}\left[\Delta\right] = \mathbb{V}\left[\tilde{R}_{KL}(\mathbf{p}; \mathbf{p}^t, \mathcal{D}_u)\right] = \frac{1}{N_u} \cdot \mathbb{V}\left[\mathbf{p}^t(x)^T(-\log(\mathbf{p}(x)))\right]$$
$$= \frac{1}{N_u} \cdot \mathbb{V}\left[\mathbf{p}^t(x)^T \log(\mathbf{p}(x))\right], \tag{38}$$

where the last equation comes from $\mathbb{V}[aX + b] = a^2 \mathbb{V}[X]$.

Finally, we plug in Eqs (37)(38) and the definition of H into Eq. (36) and complete the proof. □

8.5 Search Time of Perturbation Coefficients

For each perturbation order, we randomly sample 100 coefficient sets from $[-1, 10]$ and find the best set that has the lowest risk deviation gap according to Eq. 20 with 1000 validation examples. The whole process takes less than two minutes on CPU with 64G memory.

---

[3]This is a common assumption defined in previous literature such as (Boucheron et al. [3], Theorem 4.1; Menon et al. [22], Proposition 2) and can be achieved easily in practice with regularization techniques.