To Cool or not to Cool?
Temperature Network Meets Large Foundation Models via DRO

Zi-Hao Qiu [1]   **Siqi Guo** [2]   **Mao Xu** [1]   **Tuo Zhao** [3]   **Lijun Zhang** [1]   **Tianbao Yang** [2]

Abstract

The temperature parameter plays a profound role during training and/or inference with large foundation models (LFMs) such as large language models (LLMs) and CLIP models. Particularly, it adjusts the logits in the softmax function in LLMs, which is crucial for next token generation, and it scales the similarities in the contrastive loss for training CLIP models. A significant question remains: *"Is it viable to learn a neural network to predict a personalized temperature of any input data for enhancing LFMs?"* In this paper, we present **a principled framework** for learning a small yet generalizable temperature prediction network (TempNet) to improve LFMs. Our solution is composed of a novel learning framework with a robust loss underpinned by constrained distributionally robust optimization (DRO), and a properly designed TempNet with theoretical inspiration. TempNet can be trained together with a large foundation model from scratch or learned separately given a pretrained foundation model. It is not only useful for predicting personalized temperature to promote the training of LFMs but also generalizable and transferable to new tasks. Our experiments on LLMs and CLIP models demonstrate that TempNet greatly improves the performance of existing solutions or models, e.g. Table 1. The code to reproduce the experimental results in this paper can be found at https://github.com/zhqiu/TempNet.

Table 1. Evaluations on Instruction Following Capabilities using AlpacaEval[2]. TT stands for training time measured by GPU hours on A100-80G GPUs. The evaluations of the TempNet-enabled models were conducted in April, 2024 and the results of other models are from Alpaca leaderboard in April, 2024. The win rate measures the fraction of time the model's output is preferred over the GPT4_turbo's outputs. Length controlled (LC) win-rates are a debiased version of the win-rates that control for the length of the outputs. More details for our training can be found in Section 6.4.

Model	TT (h)	LC Win Rate	Win Rate
LLaMA2 7B (leaderboard)	−	5.35%	4.96%
LLaMA2 7B w/ TempNet	1.42	**5.74**%	**5.43**%
LLaMA2 13B (leaderboard)	−	8.43%	7.70%
LLaMA2 13B w/ TempNet	3.68	**8.58**%	**7.73**%
LLaMA2 70B (leaderboard)	−	14.69%	13.89%
LLaMA2 70B w/ TempNet	11.4	**15.83**%	**15.05**%

1 Introduction

Originating from statistical mechanics (Jaynes, 1957), temperature scaling is widely applied in softmax-type functions for training and/or inference with LFMs such as LLMs and multi-modal CLIP models[1]. In the deployment of LLMs for generating responses to prompts, a temperature-scaled softmax function models the probability distribution across a vocabulary of tokens. The CLIP model (Radford et al., 2021) is trained by optimizing a temperature-scaled contrastive loss (Oord et al., 2018) on immense scale image-text data, which has been used in many downstream tasks.

Temperature scaling plays a critical role in softmax-type functions, as increasing the temperature leads to more uniform probabilities, while decreasing it results in more concentrated probabilities. This has a notable impact in generative tasks using LLMs as varying the temperature parameter in the softmax function affects the diversity of the generated texts. For example, creative questions would benefit from higher temperatures for diverse answers, while factual questions require lower temperatures for more homogeneous responses avoiding hallucinations (Huang et al., 2023). In training CLIP models, the temperature parameter in a contrastive loss controls the degree of penalization on negative pairs, which also has a profound impact on the learned rep-

---

[1]A contrastive loss is also regarded as a softmax-type function.
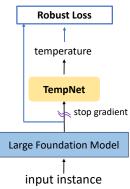[2]https://tatsu-lab.github.io/alpaca_eval/

Figure 1. Framework of Training LFMs with TempNet.

resentations. Data with frequent semantics benefit from using high temperatures for increasing tolerance of many false negatives with similar semantics, while data with rare semantics need low temperatures to enhance distinctiveness from other samples (Qiu et al., 2023; Kukleva et al., 2023).

Therefore, a significant question is *how to set the temperature parameter during the training and/or inference.* The current practice for LLMs during inference is to set it empirically depending on tasks, e.g., instruction following tasks usually use 0.7 as the default temperature (Li et al., 2023a) while math reasoning tasks usually use 0.2 as the default value. If the temperature parameter is used in training (e.g., CLIP training), a naive approach is to treat it as a hyperparameter and tune it according to some performance criteria. However, this approach is problematic as: (i) tuning the temperature parameter for training LFMs is unrealistic due to the huge cost of training; (ii) a single temperature ignores the diversity and heterogeneity of data in the real world. Hence, **an approach to automatically set the temperature based on the context/semantics is highly desirable.**

Although heuristic approaches have been adopted to tackle these issues (Radford et al., 2021; Kukleva et al., 2023; Zhang et al., 2018; Hu et al., 2017; Li et al., 2023b; Wang et al., 2020; Zhang et al., 2021; Manna et al., 2023), a more principled approach is needed. A *recent advancement* (Qiu et al., 2023) presents a novel method for contrastive learning (CL) with automatic temperature individualization. The authors presented a KL-divergence-constrained distributionally robust optimization (KL-DRO) based robust contrastive loss, incorporating an individualized temperature parameter for each anchor data optimized via Lagrangian duality theory. Although this method has proven to be effective in certain CL tasks, maintaining individualized temperatures presents several challenges. It lacks scalability for large datasets due to linearly increasing memory costs and difficulties in fully optimizing individualized parameters. Additionally, this approach poses a risk of overfitting, thereby sacrificing the method's generalizability and transferability.

While a potential solution is to utilize a neural network for predicting a personalized temperature for each sample, effec-

tively and systematically learning a temperature prediction network for multiple tasks remains a formidable challenge. In this paper, we propose a principled learning framework to learn a small yet generalizable temperature network (called TempNet) on top of LFMs, as depicted in Figure 1, which integrates a DRO-based robust loss and a properly designed TempNet. Our main contributions are:

- We propose a new framework of training LFMs by optimizing a KL-DRO based robust loss with a temperature prediction network motivated from variational analysis.

- We design novel TempNets that predict a temperature based on the output of LFMs through several layers, consisting of transformation, projection, a theory inspired parameterized pooling and an output layer.

- We conduct extensive experiments to demonstrate the effectiveness of TempNets for learning LLMs and CLIP models in various settings, including training from scratch, finetuning LLMs, and learning TempNet only with a frozen foundation model.

- We conduct deep empirical analysis of the generalizability of TempNets, the characteristics of predicted temperatures for different texts, and ablation studies of network design.

The **noticable benefits of our approach** are the following: (i) The TempNet can be efficiently trained for any LLMs with the foundation model fixed (cf. Table 1), and it only has marginal overhead when trained together with large foundation models (cf. Table 14); (ii) The predicted temperature by TempNet is context/semantics dependent. Hence, it enables the adoption of personalized temperature during the inference of LLMs such that the model knows when it should be creative by using a relatively large temperature (cf. the example shown in Figure 2) and knows when it should be more factual by using a relatively low temperature (cf. the example shown in Appendix D.7).

2 Related Work

Temperature scaling. Temperature scaling in a softmax-type function plays a significant role, which has been observed and analyzed in multiple learning problems, e.g., knowledge distillation (Hinton et al., 2015), model calibration (Guo et al., 2017), reinforcement learning (Ma et al., 2017), contrastive learning (Wang & Liu, 2021), multi-class classification with noisy labels (Zhu et al., 2023), language modeling (Wang et al., 2020; Austin et al., 2021; Chen et al., 2021; Xu et al., 2022; Gloeckle et al., 2023). Due to the critical importance of temperature scaling in softmax-type functions, multiple strategies for its modulation have been introduced, as reviewed below.

Adjusting a global temperature. A naive approach for adjusting the temperature parameter is to treat it like a hyperparameter, whose value is tuned according to some performance criteria (Hinton et al., 2015; Guo et al., 2017; Ma

Instruction: Provide a name for the dish given the ingredients and instructions.
INGREDIENTS: 2 (5 oz) cans Bumble Bee® Solid White Albacore Tuna, drained 1 avocado
2 Tbsp Sriracha
1 Tbsp Dijon mustard
2 to 3 Tbsp celery, chopped
2 Tbsp red onion, chopped
2 green onions, chopped
1 Tbsp fresh cilantro, chopped
Salt and pepper, to taste
2 heaping cups leafy green lettuce
1 cup matchstick carrots
4 (10 inch) whole wheat tortillas
INSTRUCTIONS: In a medium bowl, mash together tuna and avocado until combined. Add in the rest of the ingredients through the salt and pepper, mixing well. To assemble, top each tortilla with a 1/2 cup leafy greens, 1/4 cup matchstick carrots and divide the tuna mixture evenly among the wraps. Tightly roll up the tortilla, slice and enjoy!

LLaMA2 7B Chat w/ TempNet (avg. $\tau = 0.82$): Great! Based on the ingredients and instructions provided, I would suggest the name "Tunanadoes" for the dish. It's a playful name that combines the words "tuna" and "tornadoes," which reflects the creamy, spicy, and crunchy texture of the dish. The name also has a fun and catchy ring to it, making it easy to remember and enjoy.

LLaMA2 7B Chat w/ default $\tau = 0.7$: Thank you for sharing the ingredients and instructions for the dish! Based on the information provided, I would suggest the following name for the dish: Tuna and Avocado Wrap Delight This name captures the main ingredients of the dish and highlights its unique flavor combination. It is also a socially unbiased and positive name, free from ...

ChatGPT (GPT3.5): How about "Spicy Tuna Avocado Roll-Ups"? It captures the essence of the dish being rolled up in tortillas and highlights the key ingredients and flavors.

Figure 2. An example of instruction following, which demonstrates that TempNet makes the LLaMA2 more creative when it should be by using personalized temperatures than using the default value of 0.7. The left side displays the instruction, and the right side shows responses generated by different models. More results are included in Appendix D.7. We demonstrate the personalized τ for TempNet's responses in Figure 10.

et al., 2017). To better adapt the temperature parameter to the dynamic training process, several papers propose some schedules to adjust the temperature value during model training (Hu et al., 2017; Zhang et al., 2018; Kukleva et al., 2023; Manna et al., 2023), e.g., a cosine schedule.

Many studies use the gradient descent method for updating a global temperature based on some loss function in different learning scenarios, e.g., reinforcement learning (Kim, 2019; He et al., 2018), contrastive learning (Radford et al., 2021; Cherti et al., 2023; Goel et al., 2022; Li et al., 2021), knowledge distillation (Liu et al., 2022). However, the loss functions in these works for computing the gradient in terms of the temperature are usually heuristic or ad-hoc. For example, Radford et al. (2021); Li et al. (2021); Goel et al. (2022); Cherti et al. (2023) simply use a mini-batch based contrastive loss to compute the gradient of τ for training vision-language models such as CLIP.

Temperature Prediction. The idea of using the target network to induce a function for predicting a personalized temperature of each instance has been explored in several works (Li et al., 2023b; Wang et al., 2020; Zhang et al., 2021; Balanya et al., 2022; Joy et al., 2023; Manna et al., 2023). Wang et al. (2020) consider language modeling tasks and propose to use a single projection layer that transforms the output representation of a neural language model into a vector of temperatures corresponding to each token in the vocabulary. Zhang et al. (2021) simply use one coordinate of the embedded vector as the temperature value in contrastive learning, but observe sacrificed performance on downstream tasks. Li et al. (2023b) study knowledge distillation and use an adversarial training approach to learn a network for predicting the temperature parameter based on the predictions of the target network. In contrast to previous methods, we propose a principled training framework rooted in DRO and variational analysis, with a novel design of TempNet, leading to better improvements on both LLMs and CLIP models.

DRO-based robust losses and optimizing individualized temperatures. To the best of our knowledge, Qiu et al. (2023) is the first work that employs KL-DRO technique to formulate a robust contrastive loss, which induces individualized temperate parameters from the Lagrangian dual theory. They proposed a tailored algorithm for solving the resulting robust contrastive loss with individualized temperatures for self-supervised learning. This DRO-based formulation was also adopted in (Wu et al., 2023a;b) without considering how to optimize individualized temperatures.

3 Preliminaries

Below, we present the softmax loss of LLMs and the global contrastive loss for training CLIP models.

We first discuss the softmax loss for LLMs. Let $\mathbf{x} =$

$(t_1, \ldots, t_m)$ denote an example with a sequence of tokens, where $t_j \in \mathcal{V} = \{v_1, \ldots, v_K\}$ denotes a token from a vocabulary of size K. The probability of $\mathbf{x}$ is modeled by $p(\mathbf{x}) = \prod_{j=1}^m p(t_j | t_1, \ldots, t_{j-1})$. Modern LLMs are trained by using a softmax function to model the probability of $p(t_j | t_1, \ldots, t_{j-1})$, i.e.,

$$p(t_j | t_1, \ldots, t_{j-1}) = \frac{\exp(h(\mathbf{w}; t_1, \ldots, t_{j-1})^\top W_{t_j})}{\sum_{k=1}^K \exp(h(\mathbf{w}; t_1, \ldots, t_{j-1})^\top W_k)},$$

where $W_1, \ldots, W_K$ denotes the token embedding vectors of that in $\mathcal{V}$, $h(\mathbf{w}; t_1, \ldots, t_{j-1})$ denotes the representation of the input sequence of tokens produced by a transformer network parameterized by $\mathbf{w}$. We abuse the notation W_{t_j} to denote the embedding vector of the token t_j. The parameters $\Theta = (\mathbf{w}, W)$ are learned by minimizing the negative log-likelihood over a set of data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i = (t_1^i, \ldots, t_{m_i}^i)$:

$$\min_\Theta -\frac{1}{n} \sum_{i=1}^n [\log p(\mathbf{x}_i) := \sum_{j=1}^{m_i} \log p(t_j^i | t_1^i, \ldots, t_{j-1}^i)]. \quad (1)$$

Next we present the contrastive loss for training CLIP models. It is notable that contrastive losses have been widely used for pretraining encoder networks in a self-supervised learning fashion on various data types. For simplicity, we just restrict our discussion to training CLIP models on image-text data. However, our approach can be adopted for traing many other models based on the contrastive loss. Let $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{t}_1), \cdots, (\mathbf{x}_n, \mathbf{t}_n)\}$ denote n image-text pairs. Let $E_I(\cdot)$ and $E_T(\cdot)$ denote the encoder for images and texts parameterized by $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively, which output normalized vector representations. A traditional approach is by minimizing a mini-batch based contrastive loss (Chen et al., 2020). However, this approach is **sensitive to batch-size** and is also problematic for us to unify the softmax loss and contrastive loss. To address this issue, we adopt the two-way temperature-scaled *global contrastive loss (GCL)* (Yuan et al., 2022), which is defined below for $(\mathbf{x}_i, \mathbf{t}_i)$:

$$\ell_i(\mathbf{w}; \mathcal{D}) = -\tau_1 \log \frac{\exp(E_I(\mathbf{x}_i)^\top E_T(\mathbf{t}_i)/\tau_1)}{\sum_{\mathbf{t} \in \mathcal{T}_i^-} \exp(E_I(\mathbf{x}_i)^\top E_T(\mathbf{t})/\tau_1)}$$
$$- \tau_2 \log \frac{\exp(E_I(\mathbf{x}_i)^\top E_T(\mathbf{t}_i)/\tau_2)}{\sum_{\mathbf{x} \in \mathcal{I}_i^-} \exp(E_I(\mathbf{x})^\top E_T(\mathbf{t}_i)/\tau_2)},$$

where τ_1, τ_2 are two temperature parameters and $\mathcal{T}_i^-$ denotes the set of all negative texts of $\mathbf{x}_i$ and $\mathcal{I}_i^-$ denotes the set of all negative images of $\mathbf{t}_i$. Thus, an objective for CL is given by averaging the above GCL over all image-text pairs.

4 A Robust Learning Framework with a Temperature Prediction Network

The proposed framework for learning a temperature prediction network builds upon robust contrastive losses with individual temperature optimization underpinned by KL-divergence-constrained DRO (Qi et al., 2023; Qiu et al., 2023). To this end, we first provide a unified framework

to formulate the robust softmax loss and the robust global contrastive loss.

4.1 DRO-based Robust Losses

For any instance $z \in \Omega$ (e.g., a sequence of tokens in LLMs and an image or a text in CLIP training), assume we have a positive logit $L_+(z)$ and a finite set of contrasting logits $L_k(z), k \in \mathcal{C}$, where $\mathcal{C}$ is a discrete set and all logits are computed based on a deep neural network parameterized by Θ. The goal of learning is to push the difference $L_k(z) - L_+(z)$ to be small for $k \in \mathcal{C}$. A DRO-based robust loss is:

$$\ell_{\text{DRO}}(z) = \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k (L_k(z) - L_+(z)) - \tau_0 \text{KL}\left(\mathbf{p}, \frac{1}{|\mathcal{C}|}\right)$$
$$s.t. \text{ KL}(\mathbf{p}, 1/|\mathcal{C}|) \leq \rho,$$

where $\Delta = \{\mathbf{p} : \sum_{k \in \mathcal{C}} p_k = 1\}$ denotes a simplex, $\tau_0 > 0$ and ρ are two hyper-parameters shared by all instances, and $\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) = \sum_k p_k \log(p_k |\mathcal{C}|)$ is the KL divergence between $\mathbf{p}$ and the uniform distribution. By using the La-grangian duality theory, one can derive a dual from of the above maximization problem, yielding a closed-form of the probabilities $p_k = \frac{\exp(L_k(z)/\tau)}{\sum_{l \in \mathcal{C}} \exp(L_l(z)/\tau)}$ and an equivalent form of the robust loss (see Appendix A):

$$\ell_{\text{DRO}}(z) = \min_{\tau \geq \tau_0} f_\Theta(z, \tau) \quad (2)$$

$$f_\Theta(z, \tau) = \tau \log\left(\frac{1}{|\mathcal{C}|} \sum_{k \in \mathcal{C}} \exp\left(\frac{L_k(z) - L_+(z)}{\tau}\right)\right) + \tau\rho$$

where τ is a (shifted) Lagrangian multiplier of the KL-divergence constraint. As a result, with a set of instances $\mathcal{S} = \{z_1, \ldots, z_n\}$, the learning problem becomes:

$$\min_\Theta \frac{1}{n} \sum_{i=1}^n \min_{\tau_i \geq \tau_0} f_\Theta(z_i, \tau_i). \quad (3)$$

Below, we discuss two instantiations of the above robust objective for LLMs and contrastive learning of CLIP models.

● **Robust Softmax Loss for LLMs.** Let $z = t_{1:j-1} = (t_1, \ldots, t_{j-1})$ be a sequence of tokens followed by a token t_j. The positive logit is the prediction score $L_+(t_{1:j-1}) = h(\mathbf{w}; t_1, \ldots, t_{j-1})^\top W_{t_j}$ corresponding to the next token t_j, and the contrasting logits are prediction scores $L_k(t_{1:j-1}) = h(\mathbf{w}; t_1, \ldots, t_{j-1})^\top W_k, k \in \mathcal{V}$ of all tokens in the vocabulary $\mathcal{V}$. Hence, optimizing a robust soft-max loss over a set of data $\mathcal{D}$ for training LLMs becomes:

$$\min_\Theta \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{m_i} \min_{\tau_{ij} \geq \tau_0} f_\Theta(t_{1:j-1}^i, \tau_{ij}). \quad (4)$$

Note that if we ignore the minimization over τ_{ij} by setting it as 1 the objective becomes equivalent to that in (1) of existing apporaches for training LLMs (Brown et al., 2020).

● **Robust Contrastive Loss for CLIP training.** Let $z_i = (\mathbf{x}_i, \mathbf{t}_i)$ be an image-text pair. For an image $\mathbf{x}_i$, we denote its positive logit by $L_+(\mathbf{x}_i) = E_I(\mathbf{x}_i)^\top E_T(\mathbf{t}_i)$ and all contrasting logits by $L_k(\mathbf{x}_i) = E_I(\mathbf{x}_i)^\top E_T(\mathbf{t}_k), \forall \mathbf{t}_k \in \mathcal{T}_i^-$. Similarly, for a text $\mathbf{t}_i$, we denote its positive logit

as $L_+(\mathbf{t}_i) = E_T(\mathbf{t}_i)^\top E_I(\mathbf{x}_i)$ and contrasting logits by $L_k(\mathbf{t}_i) = E_T(\mathbf{t}_i)^\top E_I(\mathbf{x}_k), \forall \mathbf{x}_k \in \mathcal{I}_i^-$. Then, we have the following problem for contrastive learning with individualized temperatures:

$$\min_\Theta \frac{1}{n} \sum_{i=1}^n \min_{\tau_{i,1}, \tau_{i,2} \geq \tau_0} f_\Theta(\mathbf{x}_i, \tau_{i,1}) + f_\Theta(\mathbf{t}_i, \tau_{i,2}). \quad (5)$$

Different from conventional softmax loss and global contrastive loss, the robust softmax loss in (4) and the robust contrastive loss in (5) introduce a set of new temperature variables, e.g., $\{\tau_{ij}\}_{i=1,\dots,n,j=1,\dots,m_i}$ in LLMs, to be optimized. Hence, a tailored optimizer needs to be developed for solving both losses. Qiu et al. (2023) have proposed a stochastic algorithm named iSogCLR to optimize the robust contrastive losses due to the challenge that the number of logits $L_k(\mathbf{x}_i), \forall \mathbf{t}_k \in \mathcal{T}_i^{-1}$, and $L_k(\mathbf{t}_i), \forall \mathbf{x}_k \in \mathcal{I}_i^{-1}$ is the same size of dataset.

Remark: For optimizing the robust softmax loss in training LLMs, since the logits $L_k(t_{1:j-1}^i), k \in \mathcal{V}$ are computed every iteration, the optimization over τ_{ij} for those sampled tokens can be easliy solved by a traditional solver, e.g., the Newton Method (cf. Appendix B).

4.2 Robust Learning with a Temperature Network

There are several limitations of optimizing individualized temperature parameters in the above robust losses, including lack of scalability, generalizability, and transferablity.

- The **scalability issue** is profound in solving the robust contrastive losses by maintaining and updating individualized temperature parameters. This is because the memory cost of maintaining individualized temperatures increases linearly as the size of the dataset, and the individualized temperature parameters may not be fully optimized due to the limits of computing resources.

- The **lack of generalizability** is caused by potential overfitting. This is because the number of variables to be optimized is of the same size of training data. Moreover, in the real-world, the training data usually contains some noise, e.g., a word may be mispelled, and an image caption does not match its content. Thus, minimizing the the difference $L_k(z) - L_+(z)$ for all k would push the wrong positive logit $L_+(z)$ to be large. As a consequence, optimizing the robust loss with individualized temperature parameter would exacerbate such effect and harm the generalization of the learned representations.

- The **lack of transferablity** can be understood in the way that the learned temperatures are only for training instances that cannot be transferred to new data, and finding the temperature for a new data requires solving an individual optimization problem, which is costly in contrastive learning and increases inference time for using LLMs.

One potential solution to get around these issues is to express τ as a function of the input instance z, i.e., $\tau(\cdot) \in \mathrm{T}$: $\Omega \to \mathbb{R}$, such that the robust learning problem becomes:

$$\min_\Theta \inf_{\tau \in \mathrm{T}} \frac{1}{n} \sum_{i=1}^n f_\Theta(z_i, \tau(z_i)). \quad (6)$$

It is noteworthy that the lemma below from variational analysis justifies the above objective in the infinite-sample case.

Lemma 4.1 (Rockafellar & Wets (2009),Theorem 14.60)**.** *Let T be a space of measurable functions from Ω to $\mathbb{R}$ that is decomposable relative to a σ-finite measure μ on the σ-algebra $\mathcal{A}$. Let $f : \Omega \times \mathbb{R} \to \mathbb{R}$ be a normal integrand. Then, as long as $\int_{z \in \Omega} f(z, \tau(z))\mu(\mathrm{d}z) \neq \infty, \forall \tau(\cdot) \in \mathrm{T}$, we have*

$$\inf_{\tau \in \mathrm{T}} \int_{z \in \Omega} f(z, \tau(z))\mu(\mathrm{d}z) = \int_{z \in \Omega} \left(\inf_{\tau \in \mathbb{R}} f(z, \tau)\right) \mu(\mathrm{d}z).$$

Moreover, as long as the above infimum is not $-\infty$, we have that $\tau' \in \arg\min_{\tau \in \mathrm{T}} \int_{z \in \Omega} f(z, \tau(z))\mu(\mathrm{d}z)$ if and only if $\tau'(z) \in \arg\min_{\tau \in \mathbb{R}} f(z, \tau)$ for μ-almost every $z \in \Omega$.

Remark: We refer the readers to Rockafellar & Wets (2009) for more exposition of definitions of decomposable space and normal integrand. In fact, we can map the extended function $\tilde{f}_\Theta(z, \tau) = f_\Theta(z, \tau) + \delta_{\tau \geq \tau_0}(z, \tau)$ into $f(z, \tau)$ in the above lemma such that the pointwise minimization can be solved by optimization over a functional space.

Because neural networks have been extensively used as function approximators (Hornik et al., 1989), exhibiting strong capabilities in function fitting and generalization. Therefore, **our proposed solution** is solving the robust losses with a temperature prediction network:

$$\min_\Theta \min_{\mathbf{w}'} \frac{1}{n} \sum_{i=1}^n f_\Theta(z_i, \tau_{\mathbf{w}'}(z_i)). \quad (7)$$

where $\mathbf{w}'$ is the parameter of the temperature prediction network $\tau_{\mathbf{w}'}(\cdot) : \Omega \to \mathbb{R}^+$. The lemma below shows that the above objective is a relaxed upper bound of that in (3).

Lemma 4.2. *If the network is designed such that $\tau_{\mathbf{w}'}(z) \geq \tau_0$, then $\frac{1}{n} \sum_{i=1}^n \min_{\tau_i \geq \tau_0} f_\Theta(z_i, \tau_i) \leq \min_{\mathbf{w}'} \frac{1}{n} \sum_{i=1}^n f_\Theta(z_i, \tau_{\mathbf{w}'}(z_i))$.*

5 TempNet Design

The design of a proper temperature prediction network presents a significant challenge due to the extensive search space of network structures. To ensure efficiency in both training and inference, certain guiding principles have been incorporated into TempNet's design. First, TempNet shares common layers of underlying LFMs such that the size of TempNet can be greatly reduced. Second, TempNet operates directly on the output of the underlying foundation models, i.e., the output logits of LLMs and the normalized embedding vector of CLIP. The benefit of doing this is that the foundation model can be updated or frozen while learning the TempNets. Third, TempNet is inductive biased so that the generalization can be improved (Mitchell, 1980). Various inductive bias techniques are used in neural network design, such as skip connections in ResNet and normaliza-

tion layers, which are ubiquitous in contemporary networks.

To begin with, we introduce a lemma that states an implicit form of the optimal $\tau \in \mathbb{R}$ that minimizes $f(z, \tau)$ in (3).

Lemma 5.1. *Given all contrasting logits $L_k(z), k \in \mathcal{C}$ for any z, then $\tau_* = \arg\min_\tau f_\Theta(z, \tau)$ satisfies:*

$$\tau_* = \frac{1}{\rho}\left[\sum_{k \in \mathcal{C}}\left(\frac{\exp\left(\frac{L_k(z)}{\tau_*}\right)}{\sum_{j \in \mathcal{C}}\exp\left(\frac{L_j(z)}{\tau_*}\right)} - \frac{1}{|\mathcal{C}|}\right)L_k(z) - b_z\right],$$

(8)

with $b_z = \tau_* \log\left(\frac{1}{|\mathcal{C}|}\sum_{k \in \mathcal{C}}\exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \mathbb{E}_{k \in \mathcal{C}} L_k(z)$.

Remark: The above lemma shows how the optimal temperature without any constraints is pooled from the logits. The first term in the bracket is a difference between attention-based pooling and average pooling of logits, and b_z is the difference between a softmax-based pooling and average pooling of logits. The proof is deferred to Appendix C.

Then, we design TempNet by multiple layers to mimic the computation in (8). The general architecture of TempNet is depicted in Figure 3. One can observe that there are three primary components, which will be elaborated below.

The transformation-projection block. The block contains a transformation layer (a standard feed-forward layer) and a projection layer (transforms the input vector by a matrix-vector product). This block serves two purposes: (i) reducing dimensionality from outputs of LFMs; (ii) extracting semantic information useful for temperature prediction.

- **For CL:** Due to that it is prohibitive to compute all logits for each image and text at each iteration, our idea is to use the transformation-projection block to compute prototypical logits with a much smaller size for images and texts separately such that a similar pooling operation as (8) can be applied. In the following, we present the discussion for processing images, noting that texts can be processed similarly. To this end, we first transform the input normalized embedding $E_I(\mathbf{x}_i) \in \mathbb{R}^{d_0}$ into another vector $\mathbf{v} = \sigma(W_1' E_I(\mathbf{x}_i) + b_1) \in \mathbb{R}^{d_1}$ by a feed-forward layer with parameters $W_1' \in \mathbb{R}^{d_1 \times d_0}, b_1 \in \mathbb{R}^{d_1}$. Then we apply a projection layer to compute normalized prototypical logits $\mathbf{u} = \bar{W}_2'^\top \mathbf{v} \in \mathbb{R}^{d_2}$, where $\bar{W}_2' = (\bar{\mathbf{w}}_{2,1}', \ldots, \bar{\mathbf{w}}_{2,d_2}')$ is a normalized parameter matrix of $W_2' = (\mathbf{w}_{2,1}', \ldots, \mathbf{w}_{2,d_2}') \in \mathbb{R}^{d_1 \times d_2}$ such that $\bar{\mathbf{w}}_{2,k}' = \mathbf{w}_{2,k}'/\|\bar{\mathbf{w}}_{2,k}'\|_2$. We interpret the column vectors in W_2' as prototypes such that $\mathbf{u} = (u_1, \ldots, u_{d_2})$ represents prototypical logits. The size of each layer's output will obey $1 \leq d_2 \leq d_1 \leq |\mathcal{C}|$.

- **For LLMs:** The difference between TempNet for LLMs and that for CL lies at how to compute prototypical logits $\mathbf{u}$. In LLMs, the logits $L_k(z), k \in \mathcal{V}$ are computed for each instance at every iteration. However, the logit
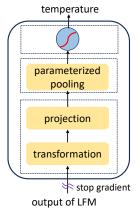

Figure 3. Structure of TempNet.

vector $L(z) = (L_1(z), \ldots, L_{d_0}(z)) \in \mathbb{R}^{d_0}$ could be high-dimensional in the same size of vocabulary $d_0 = |\mathcal{V}|$ (e.g., 32000 in LLaMA2). Hence, we first apply a transformation to a normalized logit vector to generate a lower dimensional vector $\bar{L}(z)$ by $\mathbf{v} = \sigma(W_1'\bar{L}(z) + b_1) \in \mathbb{R}^{d_1}$, where $\bar{L}(z) = L(z)/\|L(z)\|_2$. Then, we apply a projection to generate $\mathbf{u} = W_2'\mathbf{v} \in \mathbb{R}^{d_2}$. Similarly, we expect the size of each layer's output to obey $d_0 \geq d_1 \geq d_2 \geq 1$.

The parameterized pooling layer. Subsequently, we employ a parameterized pooling method to convert $\mathbf{u}$ to s mimicking (8):

$$s = \frac{1}{\rho}\left[\sum_{k=1}^{d_2}\left(\frac{\exp\left(\frac{u_k}{\phi}\right)}{\sum_{k=1}^{d_2}\exp\left(\frac{u_k}{\phi}\right)} - \frac{1}{|d_2|}\right)w_{3,k}'u_k - b\right]$$

where $\mathbf{w}_3' = (w_{3,1}', \ldots, w_{3,d_2}')$ and b, ϕ are learnable parameters, respectively. It is notable that the above pooling operation is *inductive-biased* similar to (8) expect that the scaled logits $L_k(z)/\tau_*, \forall k$ are replaced by scaled prototypical logits $u_k/\phi, k = 1, \ldots, d_2$. Our ablation studies demonstrate the effectiveness of this layer.

The output layer. At the end of the TempNet, the output layer converts s into a proper range $[\tau_0, \tau_{\max}]$ by $\tau = (\tau_{\max} - \tau_0)\,\sigma(s) + \tau_0$, where $\sigma(\cdot)$ is a sigmoid function.

Finally, we introduce the training algorithm of our proposed framework. The TempNet is learned by backpropagation. We apply stop-gradient to the input of the TempNet such that it does not affect the training dynamics of the foundation models except for the impact of the temperature parameter. The TempNet can be trained in three different ways: (i) joint training of TempNet and the foundation model from scratch; (ii) joint training of TempNet and the foundation model from a pretrained foundation model; (iii) training TempNet separately with a given foundation model. In all scenarios, the AdamW optimizer (Loshchilov & Hutter, 2017) with a consine learning rate schedule is used to update the involved network parameters. The initialization of TempNet parameters is discussed in Appendix D.1.

Table 2. Results of training LLMs in various settings, including training from scratch, finetuning a pretrained LLM model, and learning TempNet only with a frozen LLM model. More results are in Appendix D.

Setting	Common Sense Reasoning (acc(%)↑)						Language Modeling (ppl ↓)	
	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average	Lambada	Wikitext
Training GPT-2 w/ $\tau = 1.0$	60.9±1.1	**26.7**±0.4	39.2±1.0	16.5±1.1	13.9±1.6	31.44	62.49±2.70	49.86
Training GPT-2 w/ TempNet	**61.1**±1.1	26.5±0.4	**40.3**±1.0	**18.1**±1.1	**15.2**±1.6	**32.24**	**60.13**±2.43	**47.32**
Finetuning GPT-2 w/ $\tau = 1.0$	64.9±1.1	31.8±0.4	47.8±1.0	18.3±1.2	14.3±1.6	35.42	35.37±0.42	28.12
Finetuning GPT-2 w/ TempNet	**65.4**±1.1	**32.1**±0.4	**49.4**±1.0	**20.1**±1.2	**15.4**±1.6	**36.48**	**34.16**±0.42	**27.06**
Fixing LLaMA2 7B w/ $\tau = 1.0$	78.0±1.0	57.0±0.5	75.5±0.9	42.7±1.4	32.8±2.1	57.20	4.01±0.08	**11.4**
Fixing LLaMA2 7B w/ TempNet	**78.8**±1.0	**61.7**±0.5	**76.9**±0.9	**45.2**±1.4	**34.6**±2.1	**59.44**	**3.21**±0.07	12.2
Fixing LLaMA2 13B w/ $\tau = 1.0$	79.5±1.0	60.2±0.5	78.7±0.8	47.0±1.5	34.6±2.1	60.00	3.62±0.07	**10.0**
Fixing LLaMA2 13B w/ TempNet	**79.7**±0.9	**62.7**±0.5	**79.5**±0.8	**48.1**±1.5	**36.2**±2.2	**61.24**	**2.97**±0.07	11.1

Table 3. Results on contrastive learning. For image-text retrieval on Flickr30K and MSCOCO, we compute IR@1 and TR@1 for the Recall@1 on image-retrieval (IR) and text-retrieval (TR). For classification tasks, we compute top-1 accuracy (%). We report the average of scores and standard deviation over 3 runs with different random seeds. More results are in Tables 7, 8, and 9 in Appendix D.

METHOD	FLICKR30K RETRIEVAL		MSCOCO RETRIEVAL		ZERO-SHOT CLASSIFICATION TOP-1 ACC		
	IR@1	TR@1	IR@1	TR@1	CIFAR10	CIFAR100	IMAGENET1K
CLIP	40.98±0.22	50.90±0.17	21.32±0.12	26.98±0.21	60.63±0.19	30.70±0.11	36.27±0.17
CYCLIP	42.46±0.13	51.70±0.23	21.58±0.19	26.18±0.24	57.19±0.20	33.11±0.14	36.75±0.21
SOGCLR	43.32±0.18	57.18±0.20	22.43±0.13	30.08±0.22	61.09±0.24	33.26±0.12	37.46±0.19
iSOGCLR	44.36±0.12	60.20±0.26	23.27±0.18	32.72±0.13	58.91±0.15	33.81±0.18	40.72±0.23
TEMPNET	**46.17**±0.14	**62.51**±0.19	**24.83**±0.16	**34.50**±0.16	**61.77**±0.18	**34.69**±0.17	**42.28**±0.19

6 Experiments

We conduct experiments on LLMs and bimodal CL to illustrate the effectiveness and provide deep analysis of the proposed approach. The details of datasets and training setup are deferred to Appendix D.2 and D.3, respectively.

6.1 Experiments on LLMs

Setup. We consider three experimental settings: (i) training a LLM from scratch with TempNet; (ii) finetuning a LLM with a TempNet; (iii) learning a TempNet with a fixed LLM. For all settings, we use the OpenWebText2 (Gao et al., 2020) dataset for training, which contains about 17M documents of 66G from Reddit. It is notable that this dataset is relatively small compared with those used for training state-of-the-art LLMs. However, we emphasize that the major goal of the experiments is a proof of concept and obtaining state-of-the-art performance by using enormous data is out of our scope because of limited available computing resources. Indeed, it is good enough for learning a good TempNet given a fixed LLM compared with using a larger dataset (cf. Table 23).

For setting (i), we train a GPT-2 model from scratch using the GPT-NeoX library (Andonian et al., 2023) with an initial learning rate 6e-4 and a total of 320k iterations. For setting (ii), we finetune a pretrained GPT-2 model from Pythia (Biderman et al., 2023) with an initial learning rate of 1e-5 and a total of 50k iterations. For setting (iii), we experiment with several LLaMA models (Touvron et al., 2023a;b), including LLaMA1-7B, LLaMA2-7B and LLaMA2-13B. The widths

of the transformation and projection layers of all TempNets are set to $d_1 = d_2 = 256$. This means that the TempNet only has 8.2M parameters. The values for τ_0 and $\tau_{\max}$ are set to 0.001 and 2.0. The parameter ρ in our robust loss and the learning rate for TempNet are tuned within the ranges of $\{9.0, 9.5, 10.0, 10.5\}$ and $\{5e\text{-}4, 1e\text{-}4, 5e\text{-}5, 1e\text{-}5\}$, respectively. The criteria for tuning these two parameters is to simply ensure the final averaged temperature value fall in $[0.7, 1.0]$, a typical range for τ in the LLMs generative tasks (Chen et al., 2021; Touvron et al., 2023b). The impact of ρ on the performance is shown in Table 15, which indicates that $\rho = 10$ is a good value yielding an average temperature value about 0.81.

Evaluation & Results. We evaluate the learned TempNet-enabled LLMs (T-LLMs) on common-sense reasoning tasks and language modeling tasks. In common sense reasoning tasks, we use multiple multi-choice Q&A datasets (cf. Appendix D.2) and report accuracy. In language modeling tasks, we use two commonly used datasets Lambda and Wikitext and report perplexity. For TempNet-enabled LLMs, we compute the perplexity using the temperature-scaled probabilities using the temperature predicted by the learned TempNet. For all evaluations, we utilize the lm-evaluation-harness library (Gao et al., 2023).

We present the results in Table 2, and have the following observations: (1) in all three settings, LLMs with TempNet have considerable improvements on almost all tasks over their variants with a fixed $\tau = 1$, except for two cases.

For example, the TempNet-enabled LLaMA2-7B model achieves 4.7% and 1.5% improvements on HellaSwag and ARC-c datasets over the baseline, respectively; (2) TempNet-enabled LLaMA2-7B is better than the original LLaMA2-13B model on HellaSwag and Lambda and is competitive on other tasks. This is a dramatic improvement considering that TempNet has only 8.2M parameters and its training cost with fixed LLaMA2-7B is about 3 GPU hours on Nvidia 4090. In Fig. 4, we visualize the training curves regarding perplexity for setting (i) and (ii). One can observe that TempNet consistently leads to more efficient and effective model optimization process. More training curves are in Fig. 7 in Appendix D.4.

6.2 Experiments on Bimodal Contrastive Learning

Setup. We conduct experiments on CC3M data (Sharma et al., 2018) for bimodal contrastive learning. Following recent studies (Li et al., 2021; Dou et al., 2022; Qiu et al., 2023), our model incorporates the use of ResNet-50 and a transformer as image and text encoders respectively, initialized with weights from unimodal pretraining as done in (Qiu et al., 2023). The training process entails a batch size of 512 over 30 epochs. The values for τ_0 and τ_{max} are 0.001 and 0.05. The widths of the transformation and projection layers of TempNet are set to $d_1 = d_2 = 256$, which means that the TempNet only has 0.13M parameters. The TempNet is learned in the optimization framework of SogCLR (Yuan et al., 2022). The parameter ρ and the learning rate of the TempNet are tuned within the ranges of $\{7.0, 8.0, 9.0, 10.0\}$ and $\{1e\text{-}4, 5e\text{-}5, 1e\text{-}5, 5e\text{-}6\}$, respectively. As for tuning these two parameters, we simply ensure that the final averaged temperature value in a range of $[0.01, 0.02]$, which is commonly used by previous methods (Radford et al., 2019; Li et al., 2021).

Evaluation & Results. Evaluation is conducted on two downstream tasks: cross-modal retrieval and zero-shot image classification, adhering to established evaluation protocols (Radford et al., 2021; Goel et al., 2022). For retrieval, we use two datasets Flickr30K (Plummer et al., 2015) and MSCOCO (Lin et al., 2014), and for zero-shot classification we use three standard image datasets namely CIFAR10, CIFAR100, and ImageNet1K. We compare with several baselines, including CLIP, CyCLIP (Goel et al., 2022), SogCLR (Yuan et al., 2022) and iSogCLR (Qiu et al., 2023), which employ different strategies for adjusting the temperature. CLIP and CyCLIP use a heuristic apporach to optimize a temperature parameter on the fly, SogCLR uses a tuned global temperature $\tau = 0.01$, and iSogCLR optimizes individualized temperature variables on the fly.

We present partial results in Table 3 and full results in Tables 7, 8, and 9 in Appendix D.4. Compared with baselines, our algorithm achieves significant improvements on both downstream tasks. Specifically, the improvement of Temp-
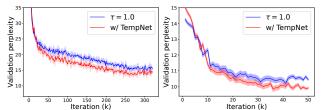

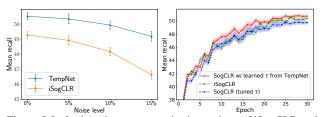*Figure 4.* Left: training GPT-2. Right: fine-tuning GPT-2.

Figure 5. Left: the robustness to noise in captions of iSogCLR and TempNet. Right: the tranferability of TempNet. (cf. the texts).

Net over iSogCLR could be as high as 2.3%. Note that TempNet and iSogCLR only differ on how the temperatures are predicted. Hence, the improvement over iSogCLR directly proves the effectiveness of using TempNet.

6.3 Generalization Abilities of TempNet

We further demonstrate the generalization abilities of TempNet for learning LLMs and contrastive learning by comparing with the individual temperature optimization under the same robust learning framework.

LLMs with TempNet vs Individually Optimized τ. As mentioned at the end of section 4.1, the individual temperature in the robust softmax loss can be directly optimized by using the Newton method. In Table 4, we refer to the method of optimizing personalized temperatures in this manner as Baseline2, and we compare it with our approach using TempNet. One can observe that employing TempNet yields better results compared to optimizing individualized temperatures, demonstrating TempNet has a better generalization capability. Additionally, in Table 4, we further verify the impact of DRO-based robust loss. Specifically, we train TempNet along with a standard temperature-scaled cross-entropy loss referred to as Baseline3. We can observe that solely using DRO loss or TempNet does not improve the standard training method (Baseline1) much. It is the combination of the DRO loss and TempNet that improves the the standard training method by a noticable margin.

Increased Robustness to Noisy Captions for CLIP training. To further demonstrate the robustness of TempNet, we consider training a CLIP model in a setting where the text caption of images are noisy, which is very common in real world. To this end, we use the CC3M dataset and manually add noise to the captions. In particular, we randomly choose $p\%$ of captions and randomly delete one word or substitute one word with 'random'. We vary the noise level $p\%$ among $\{5\%, 10\%, 15\%\}$. We report the mean recall on MSCOCO data of models learned by our method and

Table 4. The relative importance of TempNet and DRO-based robust loss for training GPT-2.

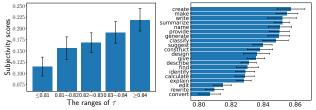| Setting	Common Sense Reasoning (acc(%)↑)						Language Modeling (ppl↓)	
	PIQA	Hellaswag	ARC-e	ARC-c	OBQA	Average	Lambada	Wikitext
Baseline1 (No TempNet, No DRO loss)	60.9±1.1	**26.7**±0.4	39.2±1.0	16.5±1.1	13.9±1.6	31.44	62.49±2.70	49.86
Baseline2 (No TempNet, DRO loss)	60.5±1.1	26.3±0.4	39.8±1.0	17.7±1.1	14.6±0.3	31.78	62.17±2.65	49.13
Baseline3 (TempNet, No DRO loss)	60.8±1.1	**26.7**±0.4	39.5±1.0	17.5±1.1	14.3±1.6	31.76	61.48±2.56	48.61
Ours (TempNet, DRO loss)	**61.1**±1.1	26.5±0.4	**40.3**±1.0	**18.1**±1.1	**15.2**±1.6	**32.24**	**60.13**±2.43	**47.32**

Figure 6. The relationship between input text characteristics (left: subjectivity, right: instruction type) and its temperature values.

iSogCLR under different noise levels in Fig. 5 (left). We can see that TempNet exhibits greater robustness to noise, hence indicating a more pronounced generalizability. Additionally, we visualize the distributions of temperature parameters generated by both algorithms under different noise levels in Fig. 8 in Appendix D.4. TempNet has an effect of correct the wrong temperatures of noisy images with frequent semantics predicted by iSogCLR.

Lastly, we show a transferability result of TempNet in CL. A TempNet is learned with a fixed CLIP model on CC12M data (Changpinyo et al., 2021). Then we use it to predict temperatures on CC3M data for contrastive learning on CC3M using SogCLR. The red curve (mean recall on MSCOCO) in Fig. 5 (right) demonstrates the effectiveness of this approach as compared with iSogCLR and SogCLR with a tuned temperature on CC3M data. Details are in Appendix D.4.

6.4 Evaluation on Instruction Following

We evaluate the instruction following ability of TempNet-enabled LLMs by using the evaluator of AlpacaEval 2.0 (Li et al., 2023a). In order to compare models on the leaderboard of AlpacaEval, we use the LLaMA2 Chat models, including the 7B, 13B, and 70B variants, which are fine-tuned versions of LLaMA2 tailored for dialogue scenarios. We train TempNets on OpenWebText2 data with fixed LLaMA2 Chat models using hyperparameter settings similar to that in Section 6.1. Specifically, $\tau_0 = 0.001, \tau_{max} = 2.0, \rho = 10$. However, as noted before the resulting TempNet gives an average temperature values around 0.81, which is larger than the default value of 0.7 for instruction following. To correct this, we use a simple trick to rescale the temperature values during the inference time by scaling τ_{max} such that the averaged predicted temperature value is around 0.7. We refer the readers to Figure 9 (right) for the impact of τ_{max} at the inference time. The results are reported in Table 1. One can see that TempNet achieves significant improvements in

generative tasks. We have also conducted more detailed case studies to analyze where the improvements come from in Appendix D.7. The results show that TempNet can output a higher temperature to enhance the flexibility of answers for open-ended questions, and a lower temperature to improve the accuracy of answers for factual questions.

6.5 Characteristics of Predicted Temperatures

Here, we analyze the characteristics of predicted temperatures by TempNet of LLMs for language data. To this end, we learn a TempNet with a fixed LLaMA1-7B model on OpenWebText2 following the setup described in section 6.1.

Texts of High Subjectivity have High Temperatures. We first conduct an experiment on the SubjQA dataset (Bjerva et al., 2020). It contains about 10k questions and answers, and they are assigned a subjectivity score by annotators. We compute the mean values of predicted temperatures by TempNet on a text prompt consisting of a question and the answer, and use it to represent the overall temperature of each question-answer pair. In Fig. 6 (left), we categorize all question-answer pairs into five distinct groups based on their temperature values and calculate the average subjectivity score for each group. The results indicate a high positive correlation between subjectivity scores and predicted temperature values of question-answer pairs.

Creative instructions have High Temperatures. We further conduct experiments on Alpaca 52k instruction following datasets (Taori et al., 2023). We calculate the average temperature values for each instruction sample using a similar method to our previous experiment. Following the procedure outlined by Wang et al. (2022), we categorize all instruction samples based their root verbs into 20 major categories. Subsequently, we compute the mean value of the temperatures for each category, and present the results in Fig. 6 (right). One can observe that the top five instruction types with the highest temperatures are relatively **subjective**, including *create*, *make*, *write*, *summarize*, and *name*. Conversely, the five instruction types with the lowest temperatures correspond to more **objective** tasks, such as *calculate*, *explain*, *edit*, *rewrite*, and *convert*.

6.6 More Studies

Due to the limited space, more studies are provided in Appendix D. Specifically, we display the training curves for different methods on GPT-2 and CLIP models in Appendix D.4, along with the performance of our method on other generative tasks such as GSM8K (Cobbe et al., 2021) and MT-bench (Zheng et al., 2024), and an analysis of time and space complexity for our method. In Appendix D.5, we present an analysis of hyperparameters of our method, including ρ and $\tau_{\max}$. In Appendix D.6, we conduct numerous ablation studies to investigate the role of TempNet and DRO loss in our proposed framework, the impact of various components within TempNet, the effect of TempNet's size, the influence of training data. Finally, in Appendix D.7, we offer several specific case studies on AlpacaEval to gain a deeper understanding of why TempNet enhances the generative performance of LLMs.

7 Conclusion

In this paper, we have proposed a principled solution to learn a properly designed temperature prediction network (TempNet) in a robust learning framework for improving large foundation models. We have conducted extensive experiments and ablations studies to demonstrate the effectiveness of our solution and our design of TempNet for large language models and contrastive learning. As a future work, we will consider adopting our framework and TempNet to train large language models on larger datasets.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Andonian, A., Anthony, Q., Biderman, S., Black, S., Gali, P., Gao, L., Hallahan, E., Levy-Kramer, J., Leahy, C., Nestler, L., Parker, K., Pieler, M., Phang, J., Purohit, S., Schoelkopf, H., Stander, D., Songz, T., Tigges, C., Thérien, B., Wang, P., and Weinbach, S. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch, 9 2023. URL https://www.github.com/eleutherai/gpt-neox.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Balanya, S. A., Maroñas, J., and Ramos, D. Adaptive temperature scaling for robust calibration of deep neural networks. *arXiv preprint arXiv:2208.00461*, 2022.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2397–2430, 2023.

Bjerva, J., Bhutani, N., Golshan, B., Tan, W.-C., and Augenstein, I. Subjqa: a dataset for subjectivity and review comprehension. *arXiv preprint arXiv:2004.14283*, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6704–6719, 2022.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3558–3568, 2021.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International conference on machine learning*, pp. 1597–1607, 2020.

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dou, Z.-Y., Kamath, A., Gan, Z., Zhang, P., Wang, J., Li, L., Liu, Z., Liu, C., LeCun, Y., Peng, N., et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9694–9705, 2022.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Gloeckle, F., Roziere, B., Hayat, A., and Synnaeve, G. Temperature-scaled large language models for lean proof-step prediction. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023.

Goel, S., Bansal, H., Bhatia, S., Rossi, R., Vinay, V., and Grover, A. Cyclip: Cyclic contrastive language-image pretraining. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6704–6719, 2022.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

He, Y.-L., Zhang, X.-L., Ao, W., and Huang, J. Z. Determining the optimal temperature parameter for softmax function in reinforcement learning. *Applied Soft Computing*, 70:80–85, 2018.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1587–1596, 2017.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

Jaynes, E. T. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Joy, T., Pinto, F., Lim, S.-N., Torr, P. H., and Dokania, P. K. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pp. 14919–14926, 2023.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the 28th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015.

Kim, S. Adaptive temperature tuning for mellowmax in deep reinforcement learning. 2019. URL https://api.semanticscholar.org/CorpusID:208980169.

Kukleva, A., Böhle, M., Schiele, B., Kuehne, H., and Rupprecht, C. Temperature schedules for self-supervised contrastive methods on long-tail data. In *the 11th International Conference on Learning Representations*, 2023.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9694–9705, 2021.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023a.

Li, Z., Li, X., Yang, L., Zhao, B., Song, R., Luo, L., Li, J., and Yang, J. Curriculum temperature for knowledge distillation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, volume 37, pp. 1504–1512, 2023b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proceedings of the 11th European Conference on Computer Vision*, pp. 740–755, 2014.

Liu, J., Liu, B., Li, H., and Liu, Y. Meta knowledge distillation. *arXiv preprint arXiv:2202.07940*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ma, X., Yin, P., Liu, J., Neubig, G., and Hovy, E. Softmax q-distribution estimation for structured prediction: A theoretical interpretation for raml. *arXiv preprint arXiv:1705.07136*, 2017.

Manna, S., Chattopadhyay, S., Dey, R., Bhattacharya, S., and Pal, U. Dystress: Dynamically scaled temperature in self-supervised contrastive learning. *arXiv preprint arXiv:2308.01140*, 2023.

Mitchell, T. M. The need for biases in learning generalizations. 1980.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2641–2649, 2015.

Qi, Q., Lyu, J., sik Chan, K., Bai, E. W., and Yang, T. Stochastic constrained dro with a complexity independent of sample size. *Transcations of Machine Learning and Research*, 2023.

Qiu, Z.-H., Hu, Q., Yuan, Z., Zhou, D., Zhang, L., and Yang, T. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 28389–28421, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021.

Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.

Rockafellar, R. T. and Wets, R. J.-B. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2556–2565, 2018.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multibillion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Sion, M. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and finetuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wang, F. and Liu, H. Understanding the behaviour of contrastive loss. In *Proceedings of the 34th IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Wang, P.-H., Hsieh, S.-I., Chang, S.-C., Chen, Y.-T., Pan, J.-Y., Wei, W., and Juan, D.-C. Contextual temperature for language modeling. *arXiv preprint arXiv:2012.13575*, 2020.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions, 2022.

Wu, J., Chen, J., Wu, J., Shi, W., Wang, X., and He, X. Understanding contrastive learning via distributionally robust optimization. *arXiv preprint arXiv:2310.11048*, 2023a.

Wu, J., Chen, J., Wu, J., Shi, W., Zhang, J., and Wang, X. Bsl: Understanding and improving softmax loss for recommendation, 2023b.

Xu, F. F., Alon, U., Neubig, G., and Hellendoorn, V. J. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pp. 1–10, 2022.

Yuan, Z., Wu, Y., Qiu, Z.-H., Du, X., Zhang, L., Zhou, D., and Yang, T. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25760–25782, 2022.

Zhang, O., Wu, M., Bayrooti, J., and Goodman, N. Temperature as uncertainty in contrastive learning. *arXiv preprint arXiv:2110.04403*, 2021.

Zhang, X., Yu, F. X., Karaman, S., Zhang, W., and Chang, S. Heated-up softmax embedding. *CoRR*, abs/1809.04157, 2018. URL http://arxiv.org/abs/1809.04157.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. volume 36, 2024.

Zhu, D., Ying, Y., and Yang, T. Label distributionally robust losses for multi-class classification: Consistency, robustness and adaptivity. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 43289–43325, 2023.

A The Derivation of the Equivalent Form of the DRO-based Robust Loss

Let us start form the DRO-based robust loss

$$\ell_{\text{DRO}}(z) = \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - \tau_0 \text{KL}\left(\mathbf{p}, \frac{1}{|\mathcal{C}|}\right)$$

$$s.t. \quad \text{KL}(\mathbf{p}, 1/|\mathcal{C}|) \leq \rho.$$

We follow Qi et al. (2023) and adopt the Lagrangian duality theory to convert $\ell_{\text{DRO}}(z)$ into its dual form. First, we have

$$\max_{\mathbf{p} \in \Delta} \min_{\lambda \geq 0} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - \tau_0 \text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \lambda(\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \rho).$$

We apply Sion's minimax theorem (Sion, 1958) and obtain

$$\min_{\lambda \geq 0} \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - \tau_0 \text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \lambda(\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \rho),$$

which is equivalent to

$$\min_{\lambda \geq 0} \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - (\tau_0 + \lambda)(\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \rho) - \tau_0 \rho.$$

Let $\tau = \tau_0 + \lambda$, then we have

$$\min_{\tau \geq \tau_0} \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - \tau(\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \rho) - \tau_0 \rho.$$

Then, the original problem is equivalent to the following problem:

$$\min_{\Theta} \min_{\tau \geq \tau_0} \max_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} p_k(L_k(z) - L_+(z)) - \tau(\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) - \rho) - \tau_0 \rho.$$

Next, we fix $\mathbf{x} = (\Theta^\top, \tau)^\top$, derive the optimal solution $\mathbf{p}^x(\mathbf{x})$ that depends on $\mathbf{x}$, and solve the inner maximization problem. To this end, we consider the following problem

$$\min_{\mathbf{p} \in \Delta} \sum_{k \in \mathcal{C}} -p_k(L_k(z) - L_+(z)) + \tau \text{KL}(\mathbf{p}, 1/|\mathcal{C}|),$$

which has the same optimal solution as our original problem. here are actually three constraints to handle, i.e., $p_k \geq 0, \forall k$, $p_k \leq 1, \forall k$ and $\sum_{k \in \mathcal{C}} p_k = 1$. Note that the constraint $p_k \geq 0, \forall k$ is enforced by the term $p_k \log(p_k)$, otherwise the above objective will be infinity. Besides, the constraint $p_k \leq 1$ is automatically satisfied due to $\sum_{k \in \mathcal{C}} p_k = 1$ and $p_k \geq 0, \forall k$. Hence, we only need to tackle the constraint $\sum_{k \in \mathcal{C}} p_k = 1$. To this end, we define the following Lagrangian function:

$$L_{\mathbf{x}}(\mathbf{p}, \mu) = \sum_{k \in \mathcal{C}} -p_k(L_k(z) - L_+(z)) + \tau\left(\log |\mathcal{C}| + \sum_{i \in \mathcal{C}} p_i \log(p_i)\right) + \mu\left(\sum_{i \in \mathcal{C}} p_i - 1\right),$$

where $\text{KL}(\mathbf{p}, 1/|\mathcal{C}|) = \log |\mathcal{C}| + \sum_{i \in \mathcal{C}} p_i \log(p_i)$, and μ is the Lagrangian multiplier for the constraint $\sum_{i \in \mathcal{C}} p_i = 1$. The optimal solution satisfy the KKT conditions:

$$-(L_k(z) - L_+(z)) + \tau(\log(p_k^*(\mathbf{x})) + 1) + \mu = 0 \quad \text{and} \quad \sum_{i \in \mathcal{C}} p_i^* = 1.$$

From the first equation, we can derive $p_k^*(\mathbf{x}) \propto \exp((L_k(z) - L_+(z))/\tau)$. Due to the second equation, we conclude that $p_k = \frac{\exp(L_k(z)/\tau)}{\sum_{l \in \mathcal{C}} \exp(L_l(z)/\tau)}$. Plugging this optimal $\mathbf{p}^*$ into the inner maximization problem over $\mathbf{p}$, we have

$$\sum_{k \in \mathcal{C}} p_k^*(\mathbf{x})(L_k(z) - L_+(z)) - \tau\left(\log \mathcal{C} + \sum_{i \in \mathcal{C}} p_i^* \log(p_i^*)\right) = \tau \log\left(\frac{1}{\mathcal{C}} \sum_{k \in \mathcal{C}} \exp\left(\frac{L_k(z) - L_+(z)}{\tau}\right)\right),$$

Therefore, we get the following equivalent problem:

$$\min_{\tau \geq \tau_0} \tau \log\left(\frac{1}{\mathcal{C}} \sum_{k \in \mathcal{C}} \exp\left(\frac{L_k(z) - L_+(z)}{\tau}\right)\right) + (\tau - \tau_0)\rho, \tag{9}$$

which is actually (2) because $\tau_0 \rho$ is a constant.

B   Newton Method for Solving the Optimal Temperature for DRO-based robust loss

First, let us recall the objective $f_\Theta(z,\tau)$:

$$f_\Theta(z,\tau) = \tau \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}}\exp\left(\frac{L_k(z) - L_+(z)}{\tau}\right)\right) + \tau\rho.$$

Then the update rules for Newton method are

$$\tau_i^0 = 1.0 \text{ (the initial guess)},$$

$$\tau_i^{n+1} = \tau_i^n - \frac{\nabla_\tau f_\Theta(z,\tau_i^n)}{\nabla_{\tau\tau} f_\Theta(z,\tau_i^n)},\ n = 0,1,\cdots$$

where $\nabla_\tau f_\Theta(z,\tau_i^n)$ and $\nabla_{\tau\tau} f_\Theta(z,\tau_i^n)$ can be computed quickly according to their definitions. Specifically, we have

$$\nabla_\tau f_\Theta(z,\tau) = \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}}\exp\left(\frac{L_k(z)}{\tau}\right)\right) - \sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\frac{L_k(z)}{\tau} + \rho, \tag{10}$$

and

$$\nabla_{\tau\tau} f_\Theta(z,\tau) = \sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left(-\frac{L_k(z)}{\tau^2}\right)$$

$$-\sum_{k\in\mathcal{C}}\left\{\nabla_\tau\left(\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\right)\frac{L_k(z)}{\tau} + \frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left(-\frac{L_k(z)}{\tau^2}\right)\right\}$$

$$= -\sum_{k\in\mathcal{C}}\nabla_\tau\left(\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\right)\frac{L_k(z)}{\tau}$$

$$= -\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left[\sum_{j\in\mathcal{C}}\frac{\exp\left(\frac{L_j(z)}{\tau}\right)}{\sum_{i\in\mathcal{C}}\exp\left(\frac{L_i(z)}{\tau}\right)}\frac{L_j(z) - L_k(z)}{\tau^2}\right]\frac{L_k(z)}{\tau}$$

$$= -\frac{1}{\tau}\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left[\sum_{j\in\mathcal{C}}\frac{\exp\left(\frac{L_j(z)}{\tau}\right)}{\sum_{i\in\mathcal{C}}\exp\left(\frac{L_i(z)}{\tau}\right)}\frac{L_j(z) - L_k(z)}{\tau}\right]\frac{L_k(z)}{\tau}.$$

Let us denote

$$A = \sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\frac{L_k(z)}{\tau}.$$

Thus, we obtain

$$\nabla_{\tau\tau} f_\Theta(z,\tau) = -\frac{1}{\tau}\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left[A - \frac{L_k(z)}{\tau}\right]\frac{L_k(z)}{\tau}$$

$$= -\frac{1}{\tau}\left[A\cdot\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\frac{L_k(z)}{\tau} - \sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left(\frac{L_k(z)}{\tau}\right)^2\right] \tag{11}$$

$$= \frac{1}{\tau}\left[\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\left(\frac{L_k(z)}{\tau}\right)^2 - \left(\sum_{k\in\mathcal{C}}\frac{\exp\left(\frac{L_k(z)}{\tau}\right)}{\sum_{j\in\mathcal{C}}\exp\left(\frac{L_j(z)}{\tau}\right)}\frac{L_k(z)}{\tau}\right)^2\right].$$

Therefore, by plugging τ_i^n in (10) and (11), we achieve the gradient and Hessian for Newton method.

C   Proof of Lemmas

We first prove Lemma 4.2.

Proof. Let τ_i^* denote the minimizer of $f_\Theta(z_i, \tau_i)$, i.e., $\tau_i^* = \arg\min_{\tau_i} f_\Theta(z_i, \tau_i)$, which gives us $f_\Theta(z_i, \tau_i^*) \leq \min_{\mathbf{w}'} f_\Theta(z_i, \tau_{\mathbf{w}'}(z_i))$. Thus, we have

$$\frac{1}{n}\sum_{i=1}^n \min_{\tau_i \geq \tau_0} f_\Theta(z_i, \tau_i) = \frac{1}{n}\sum_{i=1}^n f_\Theta(z_i, \tau_i^*) \leq \frac{1}{n}\sum_{i=1}^n \min_{\mathbf{w}'} f_\Theta(z_i, \tau_{\mathbf{w}'}(z_i)) = \min_{\mathbf{w}'} \frac{1}{n}\sum_{i=1}^n f_\Theta(z_i, \tau_{\mathbf{w}'}(z_i)).$$

$\square$

Then we aim to prove Lemma 5.1.

Proof. Recall that

$$f_\Theta(z, \tau) = \tau \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z) - L_+(z)}{\tau}\right)\right) + \tau\rho$$

As $\tau_* = \arg\min_\tau f_\Theta(z, \tau)$, we have $\nabla_\tau f_\Theta(z, \tau_*) = 0$, i.e.,

$$\log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z) - L_+(z)}{\tau_*}\right)\right) - \sum_{k\in\mathcal{C}} \frac{\exp\left(\frac{L_k(z)-L_+(z)}{\tau_*}\right)}{\sum_{j\in\mathcal{C}} \exp\left(\frac{L_j(z)-L_+(z)}{\tau_*}\right)} \frac{L_k(z) - L_+(z)}{\tau_*} + \rho = 0,$$

which can be simplified into

$$\log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \sum_{k\in\mathcal{C}} \frac{\exp\left(\frac{L_k(z)}{\tau_*}\right)}{\sum_{j\in\mathcal{C}} \exp\left(\frac{L_j(z)}{\tau_*}\right)} \frac{L_k(z)}{\tau_*} + \rho = 0,$$

and we have

$$\tau_* \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \sum_{k\in\mathcal{C}} \frac{\exp\left(\frac{L_k(z)}{\tau_*}\right)}{\sum_{j\in\mathcal{C}} \exp\left(\frac{L_j(z)}{\tau_*}\right)} L_k(z) + \rho\tau_* = 0.$$

Let $b_z = \tau_* \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \mathbb{E}_{k\in\mathcal{C}} L_k(z)$, and we obtain

$$b_z + \mathbb{E}_{k\in\mathcal{C}} L_k(z) - \sum_{k\in\mathcal{C}} \frac{\exp\left(\frac{L_k(z)}{\tau_*}\right)}{\sum_{j\in\mathcal{C}} \exp\left(\frac{L_j(z)}{\tau_*}\right)} L_k(z) + \rho\tau_* = 0.$$

By rearranging the terms in the above equation, we have

$$\tau_* = \frac{1}{\rho}\left[\sum_{k\in\mathcal{C}} \frac{\exp\left(\frac{L_k(z)}{\tau_*}\right)}{\sum_{j\in\mathcal{C}} \exp\left(\frac{L_j(z)}{\tau_*}\right)} L_k(z) - \mathbb{E}_{k\in\mathcal{C}} L_k(z) - b_z\right].$$

Moreover, we can bound b_z as follows. By using the properties of *logsumexp* function, we have

$$b_z = \tau_* \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \mathbb{E}_{k\in\mathcal{C}} L_k(z) = \tau_* \log\left(\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \log|\mathcal{C}| - \mathbb{E}_{k\in\mathcal{C}} L_k(z)$$

$$\leq \tau_* \log\left(\sum_{k\in\mathcal{C}} \exp\left(\frac{\max_{k\in\mathcal{C}}\{L_k(z)\}}{\tau_*}\right)\right) - \log|\mathcal{C}| - \mathbb{E}_{k\in\mathcal{C}} L_k(z) = \max_{k\in\mathcal{C}}\{L_k(z)\} - \mathbb{E}_{k\in\mathcal{C}} L_k(z),$$

and due to e^x is convex w.r.t. x and $\mathbb{E}[e^x] \geq e^{\mathbb{E}[x]}$, we have

$$b_z = \tau_* \log\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \exp\left(\frac{L_k(z)}{\tau_*}\right)\right) - \mathbb{E}_{k\in\mathcal{C}} L_k(z) \geq \tau_* \log\exp\left(\frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \frac{L_k(z)}{\tau_*}\right) - \mathbb{E}_{k\in\mathcal{C}} L_k(z)$$

$$= \tau_* \frac{1}{|\mathcal{C}|}\sum_{k\in\mathcal{C}} \frac{L_k(z)}{\tau_*} - \mathbb{E}_{k\in\mathcal{C}} L_k(z) = 0.$$

$\square$

Table 5. Hyper-parameters for each LLMs experiment group. *We use 20k iterations to train LLaMA2 Chat 70B.

Hyper-parameters	Training GPT-2	Finetuning GPT-2	Fixing LLaMA1	Fixing LLaMA2	Fixing LLaMA2 Chat
Warmup Steps Percentage	0.01	0.01	0.01	0.01	0.01
Init Learning Rate	6e-4	1e-4	2e-4	1e-4	1e-4
Batch Size	32	32	16	16	16
Weight Decay	0.1	0.1	0.1	0.01	0.01
Training Iterations	320k	50k	10k	30k	30k*
Learning Rate Decay	Cosine	Cosine	Cosine	Cosine	Cosine
Adam ϵ	1e-8	1e-8	1e-8	1e-8	1e-8
Adam β_1	0.9	0.9	0.9	0.9	0.9
Adam β_2	0.95	0.95	0.95	0.999	0.999

D Experiments

D.1 Details of Implementation

Initialization. For the parameters of the transformation layer in TempNet, we employ the widely used kaiming uniform initialization (He et al., 2015) in neural networks. The projection layer in TempNet is designed to generate prototypical logits, hence in the context of contrastive learning tasks, we utilize the initial representations of randomly sampled images/texts to initialize the text/image TempNet's projection layer. For the TempNet associated with LLMs, the projection layer is initialized using the kaiming uniform method. The parameter $\mathbf{w}_3'$ in the parameterized pooling, intended to autonomously weigh different components in the prototypical logits, is initialized as an all-ones vector. We set ϕ to the temperature values commonly used in specific tasks, such as 1.0 for LLMs and 0.01 for CLIP models. Additionally, we initialize b to 0.

Codebase. In the experiments of LLMs, we use multiple code frameworks. For the GPT-2 experiments, we employed the GPT-NeoX framework (Andonian et al., 2023), which leverages good features as the popular Megatron-DeepSpeed (Rasley et al., 2020) library but with substantially increased usability, e.g., model evaluation. In the fine-tuning experiments of GPT-2, we used pretrained weights from Pythia (Biderman et al., 2023). For the LLaMA experiments, we utilized the Megatron framework (Shoeybi et al., 2019), and for the LoRA experiments, we used the alpaca-lora framework[2]. When testing model performance, we primarily used the lm-evaluation-harness library (Gao et al., 2023), which provides numerous common datasets and related tasks. For bimodal contrastive learning experiments, we adopt the code base from (Qiu et al., 2023). We also adopt ResNet-50 as the image encoder and DistilBert (Sanh et al., 2019) as the text encoder.

D.2 Details of Datasets

LLMs Experiments. We mainly use OpenWebText2 dataset (Gao et al., 2020), which is part of EleutherAi/The Pile dataset and is an enhanced version of the original OpenWebTextCorpus covering all Reddit submissions from 2005 up until April 2020. During evaluation, we utilized two datasets Lambada and Wikitext-2 for assessing Language Modeling performance, and five question-answer datasets, PIQA, HellaSwag, ARC-e, ARC-c, OBQA, which are widely used for testing Large Language Models (LLMs) (Brown et al., 2022; Touvron et al., 2023b; Biderman et al., 2023).

Contrastive Learning Experiments. We mainly use the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018), which contains about 2.9 million image-caption pairs crawled from the Internet. Note that as time goes by, some images are not available. Thus the number of image-caption pairs we use in our experiments is smaller than that in the original papers. We also use the CC12M dataset (Changpinyo et al., 2021), which is larger and covers a much more diverse set of visual concepts than CC3M. Each image in MSCOCO and Flickr30K datasets has about 5 captions. MSCOCO dataset (Lin et al., 2014) contains 113K images and 567K captions, and Flickr30K dataset (Plummer et al., 2015) has 32K images and 158K captions. We employ the well-known Karpathy split (Karpathy & Fei-Fei, 2015) for these two datasets.

D.3 Training Setup Details

LLMs Experiments. In our experiments, we employed the GPT-2 model (125M) (Radford et al., 2019) along with a series of LLaMA models, including LLaMA1 7B (Touvron et al., 2023a), LLaMA2 7B, 13B, and versions of LLaMA2 fine-tuned for conversational tasks, namely LLaMA2 Chat 7B, 13B, 70B (Touvron et al., 2023b). GPT-2 (125M) is a 12-layer BERT-based model, and has a hidden dimension of 768. The number of layers and hidden dimensions for the LLaMA 7B, 13B, 70B models are respectively 32, 40, 80 and 4096, 5120, 8192, respectively. The training hyper-parameter

---

[2]https://github.com/tloen/alpaca-lora

Table 6. Hyper-parameters for each contrastive learning experiment group.

Hyper-parameters	CLIP Model Training	Transferability of TempNet	Robustness to Noisy Captions
Warmup Steps	1000	1000	1000
Init Learning Rate	2e-4	2e-4	2e-4
Batch Size	512	512	512
Weight Decay	0.02	0.02	0.02
Training Epochs	30	10	30
Learning Rate Decay	Cosine	Cosine	Cosine
Adam ϵ	1e-8	1e-8	1e-8
Adam β_1	0.9	0.9	0.9
Adam β_2	0.999	0.999	0.999

configurations for all experiment groups are listed in Table 5. We employ GTX 3090, 4090, A6000, and A100 GPUs in our experiments, and they have 24GB, 24Gb, 48GB, 80GB memory, respectively. The experiments on LLaMA2 Chat model (results in Table 1) are performed on A100 GPUs.

Contrastive Learning Experiments. In table 6, we report the training hyper-parameter configurations for all three contrastive learning experiment groups: training CLIP model, verifying the transferability of TempNet, and demonstration the robustness of TempNet to noisy captions. We train our models on Nvidia Tesla V100 GPU with 32 GB memory and GTX 3090 GPU with 24GB memory. It takes about 8 hours to finish 30 epochs training.
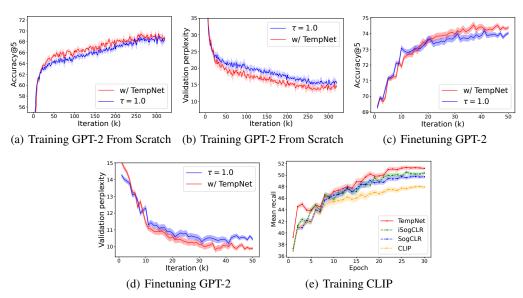
D.4 Additional Experimental Results

(a) Training GPT-2 From Scratch (b) Training GPT-2 From Scratch (c) Finetuning GPT-2

(d) Finetuning GPT-2 (e) Training CLIP

Figure 7. The training curves for the GPT-2 model (a-d), and for the CLIP model (e).

Explanation of Performance Improvement on Common Sense Reasoning Tasks. Given a question $\mathbf{x}_0$ and an text of answer option $\mathbf{x}_1$, the lm-evaluation-harness library (Gao et al., 2023) calculates **the log likelihood** of $\log p(\mathbf{x}_1|\mathbf{x}_0)$. The answer option with the highest log likelihood is returned as the answer.

Training Curves. We demonstrate the training curves of different methods for GPT-2 model and CLIP in Figure 7. It is notable that TempNet actually promotes the training of both language models and CLIP models.

More Experimental Results on LLaMA1 7B. In Table 10, we demonstrate the significant improvement achieved by TempNet on the LLaMA1 7B model.

More Experimental Results on Bimodal CL. We show full results of zero-shot retrieval results on MSCOCO and Flick30K, and zero-shot classification accuracy on CIFAR-10, CIFAR-100, and ImageNet1K in Table 7, 8, and 9, respectively.

Visualization of the Learned Temperature Distributions. To further investigate why TempNet outperforms iSogCLR, we compare the distributions of τ learned by these methods at different noise levels in the Figure 8. We first set the noise level to 0% (i.e., no noise added to the data) and showcase the scenarios for iSogCLR and TempNet in Figure 8(a) and (c), respectively. In these two figures, the blue part represents the distribution of tau for all images, the orange part shows the distribution for 15% of the images (we selected samples with IDs in the top 15%, to which we will add noise later), and the red part indicates the top 20% of the orange samples with the highest temperature values, which can be considered as samples with frequent semantics. Subsequently, we set the noise level to 15%, adding noise to the samples with IDs in the top 15%, and display the distribution of temperature values for iSogCLR and TempNet in (b) and (d), respectively. As before, the blue section represents all samples, the orange section denotes the 15% of samples to which noise was added, and the red section highlights those among the orange with frequent semantics. It is evident that in iSogCLR, the samples represented in red almost exclusively learn a very small temperature values, indicating that their temperatures are underestimated due to noise. In contrast, TempNet is able to correct these samples' temperature values, ensuring they were not too low. Hence, this demonstrates that TempNet is more robust to noise when learning temperature values compared to iSogCLR.

Transferability of TempNet in Contrastive Learning. We conduct an experiment in a transfer learning setting to verify the generalizability of TempNet for contrastive learning. First, we fix a pretrained CLIP model and only learn a TempNet on the CC12M dataset (Changpinyo et al., 2021). The TempNet is trained for 10 epochs with an initial learning rate of 1e-4. Subsequently, the learned TempNet is employed to predict the temperatures of all samples in the CC3M dataset. These temperatures are then used in the SogCLR algorithm for model training on CC3M. We compare the resulting model against those trained by iSogCLR and SogCLR (with a tuned temperature 0.01) on the same dataset, and report the mean recall results on MSCOCO data in Fig. 5 (left). The mean recall is computed by averaging the recalls of image retrieval and text retrieval at top 1, 5, 10 positions. As seen, using the temperatures predicted by a TempNet learned on CC12M exhibited
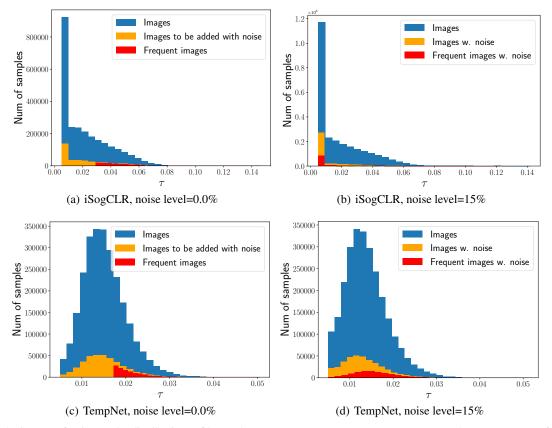
(a) iSogCLR, noise level=0.0%

(b) iSogCLR, noise level=15%

(c) TempNet, noise level=0.0%

(d) TempNet, noise level=15%

Figure 8. The impact of noise on the distributions of learned temperature parameters. TempNet corrects the temperatures of noisy data with frequent semantics, which are predicted as low by iSogCLR.

superior performance on the CC3M dataset, indicating generalization capabilities of TempNet.

Performance of TempNet on other large language model generation tasks. Below are the results on the GSM8K arithmetic reasoning task. We employ chain-of-thought prompting and the self-consistency decoding strategy. In the Table 11, maj@k denotes sampling reasoning paths and performing a majority vote over the final answer set. The baseline uses an empirically tuned value of 0.2, which is widely adopted in previous studies. Table 12 presents the comparison results of using MT-bench's default temperature values versus those generated by TempNet on three LLaMA2 Chat models. In general, the results above indicate that employing TempNet can lead to performance enhancements in different language model generation tasks.

Complexity analysis of the proposed method. We analyze the complexity of our method from both space and time perspectives:

Space complexity: Assuming the input dimension of TempNet is d_0, the dimensions after the transformation layer and projection layer are d_1 and d_2, respectively, then the total parameter count in TempNet is $d_0 \times d_1 + d_1 + d_1 \times d_2 + d_2 + 2$, with $d_0 \times d_1 + d_1$ for the transformation layer, $d_1 \times d_2$ for the projection layer, and $d_2 + 2$ or the parameterized pooling layer. We present the numbers of parameters for both the base models and their corresponding TempNets across three different settings, along with the percentages of TempNet parameters in Table 13.

Time complexity: Note that the per-iteration complexity of all gradient-based stochastic algorithms is $\mathcal{O}(Bd)$, where B is the mini-batch size and d is the number of model parameters. Therefore, the additional time overhead introduced by using TempNet is directly proportional to the number of TempNet parameters, which is significantly less than that of the base foundation models. In Table 14, we compare our method (DRO-based robust loss + TempNet) with baseline methods (standard cross-entropy loss / contrastive loss) with different base models in terms of training and inference times. The training time represents the duration to train the model for 10,000 iterations on 8 A6000 GPUs. Regarding model inference performance for language models, we measure it by throughput (tokens/s) on 1 A6000 GPU, which is the number of tokens processed per second. Note that in the experiments with the CLIP model, TempNet is not required during inference for

Table 7. Zero-shot image-text retrieval (text-to-image and image-to-text) results (Recall@k), where $k \in \{1, 5, 10\}$, on Flickr30K dataset.

METHOD	IMAGE RETRIEVAL			TEXT RETRIEVAL		
	R@1	R@5	R@10	R@1	R@5	R@10
CLIP	40.98±0.22	69.60±0.19	79.22±0.08	50.90±0.17	81.00±0.16	87.90±0.22
CYCLIP	42.46±0.13	69.56±0.16	78.74±0.21	51.70±0.23	79.90±0.18	88.40±0.11
SOGCLR	43.32±0.18	71.06±0.13	79.54±0.19	57.18±0.20	81.03±0.26	88.62±0.18
ISOGCLR	44.36±0.12	72.64±0.17	80.92±0.13	60.20±0.26	84.60±0.21	90.50±0.14
TEMPNET	**46.17**±0.14	**73.68**±0.12	**82.45**±0.15	**62.51**±0.19	**85.31**±0.20	**92.05**±0.13

Table 8. Zero-shot image-text retrieval (text-to-image and image-to-text) results (Recall@k), where $k \in \{1, 5, 10\}$, on MSCOCO dataset.

METHOD	IMAGE RETRIEVAL			TEXT RETRIEVAL		
	R@1	R@5	R@10	R@1	R@5	R@10
CLIP	21.32±0.12	45.52±0.17	57.30±0.16	26.98±0.21	54.86±0.15	66.86±0.19
CYCLIP	21.58±0.19	45.46±0.13	57.56±0.22	26.18±0.24	53.24±0.18	65.86±0.22
SOGCLR	22.43±0.13	46.74±0.11	58.32±0.20	30.08±0.22	56.94±0.17	67.39±0.24
ISOGCLR	23.27±0.18	47.23±0.24	59.07±0.19	32.72±0.13	59.52±0.11	70.78±0.21
TEMPNET	**24.83**±0.16	**49.29**±0.21	**61.05**±0.18	**34.50**±0.16	**61.26**±0.14	**72.14**±0.16

Table 9. Zero-shot top-k classification accuracy (%), where $k \in \{1, 3, 5\}$.

METHOD	CIFAR10			CIFAR100		
	TOP-1	TOP-3	TOP-5	TOP-1	TOP-3	TOP-5
CLIP	60.63±0.19	87.29±0.12	95.02±0.16	30.70±0.11	49.49±0.13	58.51±0.14
CYCLIP	57.19±0.20	85.02±0.14	93.94±0.23	33.11±0.14	52.99±0.17	61.01±0.22
SOGCLR	61.09±0.24	88.12±0.19	94.92±0.18	33.26±0.12	52.46±0.22	60.71±0.15
ISOGCLR	58.91±0.15	86.27±0.24	93.43±0.11	33.81±0.18	53.21±0.21	61.83±0.19
TEMPNET	**61.77**±0.18	**88.24**±0.21	**95.19**±0.13	**34.69**±0.17	**54.38**±0.14	**62.51**±0.15

METHOD	IMAGENET1K		
	TOP-1	TOP-3	TOP-5
CLIP	36.27±0.17	51.03±0.17	56.84±0.22
CYCLIP	36.75±0.21	51.32±0.18	57.08±0.23
SOGCLR	37.46±0.19	52.68±0.16	58.04±0.10
ISOGCLR	40.72±0.23	54.38±0.14	59.11±0.17
TEMPNET	**42.28**±0.19	**56.19**±0.17	**61.32**±0.16

downstream tasks.

One can observe that our method, compared to the standard methods using the cross-entropy loss or the contrastive loss, has slightly increased training time and slightly decreased throughput. Additionally, experiments with LLaMA models show that the larger the LLaMA model, the smaller the impact of TempNet on throughput.

Table 10. We demonstrate the improvements of TempNet on the LLaMA1 7B model, reporting the perplexity across three language modeling tasks and the accuracy on seven common sense reasoning tasks.

Setting	Common Sense Reasoning (acc(%)↑)					Language Modeling (ppl↓)	
	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Lambada	Wikitext
Fixing LLaMA1 7B $\tau = 1.0$	78.1±0.9	56.2±0.5	75.1±0.8	40.2±1.4	32.2±2.1	4.41±0.09	**9.41**
Fixing LLaMA1 7B w/ TempNet	**79.1**±0.9	**58.3**±0.4	**75.8**±0.8	**42.1**±1.4	**34.8**±2.1	**3.65**±0.08	9.87

D.5 Hyper-parameter Analysis

The influence of the hyper-parameter ρ. To demonstrate the impact of ρ, we train TempNet on the LLaMA 7B model using various values of ρ and evaluate it across 11 common sense reasoning tasks (PIQA, HellaSwag, ARC-e, ARC-c, OBQA, MathQA, WinoGrande, SciQ, BoolQ, Swag, LogiQA). In the Table 15, we show the average learned temperature parameters and the average accuracy across the aforementioned 11 tasks for different values of ρ.

One can observe that the larger the value of ρ, the smaller the average learned τ. Furthermore, our model exhibits consistently good performance within the range of $[9.5, 10.5]$ for ρ, with the overall best performance occurring at $\rho = 10.0$.

Additionally, we demonstrate in Table 16 the importance of the linear term $\rho\tau$ in our robust loss for improving the performance. Earlier works (Radford et al., 2021; Wang et al., 2020) used heuristic approaches which ignore such a term.

The impact of $\tau_{\mathbf{max}}$ in TempNet. We verify the effects of different $\tau_{\max}$ values on LLaMA, as shown in the Table 17. The results indicate the choice of has little impact on the TempNet's performance.

The influence of $\tau_{\mathbf{max}}$ during evaluation. In large language model experiments, the $\tau_{\max}$ is set to 2.0 during the training phase, but in the evaluation phase, we can set different $\tau_{\max}$ values (which we refer to as $\tau_{\max}^{\text{eval}}$). In Figure 9, we investigate the impact of $\tau_{\max}^{\text{eval}}$ on the performance of common sense reasoning tasks and AlpacaEval task. One can observe that, on both tasks, when $\tau_{\max}^{\text{eval}}$ is in the range of 1.2 to 1.4, the average τ is approximately between 0.7 and 0.8, which yields the best performance.
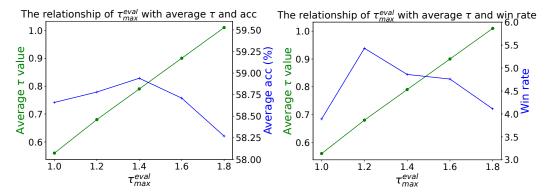

Figure 9. When $\tau_{\max}^{\text{eval}}$ takes different values, the average value of τ and the average accuracy on common sense reasoning tasks (left), and the win rate on AlpacaEval (right).

D.6 More Ablation Studies

The effects of different components in TempNet. We first investigate the effects of different components within TempNet through a series of ablation studies. The results are presented in Table 19 and 20. First, we observe that the transformation layer and projection layer lead to performance improvements, which substantiates the efficacy of these two structures. Besides, in the experiments with the language model, we discern the significance of normalizing the raw logits inputted into the TempNet. Lastly, to further study the role of parameterized pooling, we substitute this layer with a simple linear layer, which actually transforms the network into a Multi-Layer Perceptron (MLP). This variant results in a diminished performance, thereby affirming the effectiveness of the proposed inductive-biased parameterized pooling.

The relative importance of DRO-based robust loss versus TempNet. We further conduct experiments to demonstrate the importance of using distributionally robust objective by comparing the performance of our method with the baselines that

Table 11. Comparison on the GSM8K dataset.

Model	maj@1	maj@8	maj@64
LLaMA2 7B w/ $\tau = 0.2$	**13.2**±0.9	15.5±1.0	16.5±1.0
LLaMA2 7B w/ TempNet	12.8±0.9	**15.6**±1.0	**17.3**±1.0

Table 12. Comparison on the MT-bench dataset.

Model	LLaMA2 Chat 7B	LLaMA2 Chat 13B	LLaMA2 Chat 70B
w/ default τ values	6.27	**6.65**	6.89
w/ TempNet	**6.44**	6.58	**6.97**

Table 13. The number of parameters in TempNet with different base models.

Base Model	Base Model #Params	TempNet #Params	TempNet Param%
CLIP model	92M	0.26M	0.28%
GPT-2	125M	12.94M	10.35%
LLaMA 7B	6746M	8.26M	0.12%

Table 14. Comparison of training/inference time.

Method	Training GPT-2: Training Time (h)	Inference Throughput with GPT-2 (tokens/s)	Training CLIP model (h)
Baseline	1.21	9655.77	2.31
Ours	1.28	8966.07	2.42

Method	Fixing LLaMA2 7B: Training Time (h)	Inference Throughput with LLaMA2 7B (tokens/s)
Baseline	—	950.94
Ours	3.98	943.79

Method	Fixing LLaMA2 13B: Training Time (h)	Inference Throughput with LLaMA2 13B (tokens/s)
Baseline	—	590.53
Ours	10.33	587.26

learn the TempNet using the standard temperature-scaled cross-entropy loss or contrastive loss (without τ before the $\log$ function). We consider training GPT-2, fixing LLaMA model, and training CLIP models, with the results (abbreviations: "DRO" indicates DRO-based robust loss, "TS-CE" indicates temperature-scaled cross-entropy loss, "CL" indicates contrastive loss, and "SG" indicates the stop-gradient operator) demonstrated in Table 18.

From these experimental results, it is clear that first, in both language model and CLIP experiments, our DRO-based robust loss achieves results superior to the standard temperature-scaled loss/contrastive loss, validating the effectiveness of DRO-based robust loss for learning the TempNet. Specifically, in our experiments with LLaMA and CLIP models, we also observe that using standard temperature-scaled loss/contrastive loss leads to TempNet generating temperatures that continually increase until reaching $\tau_{\max}$, indicating that this approach does not effectively learn personalized temperatures in these scenarios. We also note that the use of stop-gradient operation also helps improve the performance.

The impact of TempNet's size on performance. In the development of TempNet, we conducted comprehensive investigations into how the number of network parameters influences performance. Our current TempNet consists of three layers L (i.e., a depth of 3), with $d_1 = d_2 = 256$ (i.e., a width of 256). We experiment with using more layers (such as adding an extra layer in the transformation-projection block) or different widths (such as 64 or 512), with the results demonstrated in Table 21 and 22. From these results, it is notable that a network with a depth of 3 and a width of 256 performs well on both tasks, and increasing the width does not significantly enhance performance. Moreover, using a network with a depth of 4 results in a slight decline in performance. We attribute this to the possibility that larger networks are more susceptible to data noise.

The impact of training datasets in large language model experiments. Below, we demonstrate the impact of training on different datasets on performance. Specifically, we opt for a comparison with the SlimPajama dataset, noting that it is larger in scale and encompasses a broader array of data sources than the OpenWebText2 dataset used in our manuscript. We conduct experiments on the LLaMA model, with the results presented in Table 23. We find that although the SlimPajama

Table 15. The influence of the hyper-parameter ρ. We train TempNet on the LLaMA 7B model with different ρ values and test it on 11 common sense reasoning tasks (PIQA, HellaSwag, ARC-e, ARC-c, OBQA, MathQA, WinoGrande, SciQ, BoolQ, Swag, LogiQA). The table below shows the average learned temperature parameters and accuracy for these tasks across various ρ values.

ρ	8.0	9.0	9.5	10.0	10.5	11.0
Avg. learned τ	1.65	1.03	0.95	0.81	0.69	0.35
Avg. acc (%)	56.66	57.34	57.77	**57.87**	57.74	56.86

Table 16. Results of training the GPT-2 model with TempNet but setting $\rho = 0.0$. During training, we observe that the average value of learned τ is monotonically increasing and then all output τ values become $\tau_{\max}$, which leads to suboptimal performance.

Setting	Common Sense Reasoning (acc(%)↑)					Language Modeling (ppl ↓)	
	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Lambada	Wikitext
Training GPT-2 w/ $\tau = 1.0$	60.9±1.1	**26.7**±0.4	39.2±1.0	16.5±1.1	13.9±1.6	62.49±2.70	49.86
Training GPT-2 w/ TempNet ($\rho = 0.0$)	60.4±1.1	26.3±0.4	39.4±1.0	16.6±1.1	14.0±1.6	62.66±2.43	49.61
Training GPT-2 w/ TempNet (tuned ρ)	**61.1**±1.1	26.5±0.4	**40.3**±1.0	**18.1**±1.1	**15.2**±1.6	**60.13**±2.43	**47.32**

dataset contains richer corpora, the performance of TempNet trained on it is comparable to that of TempNet trained on OpenWebText2.

Table 17. The impact of $\tau_{\max}$ in TempNet.

Fixing LLaMA1 7B	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average
$\tau_{\max} = 2.0$	**79.1**±0.9	58.3±0.4	**75.8**±0.8	42.1±1.4	**34.8**±2.1	**58.02**
$\tau_{\max} = 1.4$	**79.1**±0.9	**59.4**±0.4	75.6±0.9	**42.5**±1.4	33.2±2.1	57.96

Table 18. The relative importance of DRO-based robust loss versus TempNet. SG means stop gradient.

Training GPT-2	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average
Baseline1 (TS-CE, w/o SG)	60.6±1.1	26.6±0.4	39.6±1.0	17.7±1.1	14.2±1.6	31.74
Baseline2 (TS-CE, w/ SG)	60.8±1.1	**26.7**±0.4	39.5±1.0	17.5±1.1	14.3±1.6	31.76
Baseline3 (DRO, w/o SG)	60.9±1.1	26.5±0.4	40.0±1.0	17.8±1.1	14.9±1.6	32.02
Ours (DRO, w/ SG)	**61.1**±1.1	26.5±0.4	**40.3**±1.0	**18.1**±1.1	**15.2**±1.6	**32.24**

Fixing LLaMA1 7B	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average
Baseline1 (TS-CE)	77.0±0.9	56.5±0.4	75.1±0.8	41.8±1.4	32.1±2.1	56.50
Ours (DRO)	**79.1**±0.9	**58.3**±0.4	**75.8**±0.8	**42.1**±1.4	**34.8**±2.1	**58.02**

METHOD	FLICKR30K RETRIEVAL		MSCOCO RETRIEVAL		ZERO-SHOT CLASSIFICATION TOP-1 ACC		
	IR@1	TR@1	IR@1	TR@1	CIFAR10	CIFAR100	IMAGENET1K
CL, w/o SG	41.49±0.19	56.33±0.17	21.57±0.13	28.83±0.09	59.52±0.11	31.91±0.14	36.93±0.22
CL, w/ SG	43.43±0.21	56.62±0.16	22.17±0.20	29.68±0.14	60.65±0.15	33.22±0.13	37.07±0.16
DRO, w/o SG	42.07±0.15	56.44±0.13	21.76±0.16	29.13±0.11	60.22±0.20	32.12±0.16	37.21±0.14
DRO, w/ SG	**46.17**±0.14	**62.51**±0.19	**24.83**±0.16	**34.50**±0.16	**61.77**±0.18	**34.69**±0.17	**42.28**±0.19

D.7 Analysis of the Instruction Following Experiment Results

In this section, we demonstrate why TempNet enhances generation performance by comparing the performance of LLaMA2 7B Chat (with the default $\tau = 0.7$) and LLaMA2 7B Chat + TempNet on the AlpacaEval dataset (Li et al., 2023a). Specifically, we employ the response files generated by LLaMA2 7B Chat and LLaMA2 7B Chat + TempNet, selecting the questions where the baselines (generated by GPT-4 Turbo) are better than the answers from LLaMA2 7B Chat but weaker than that from LLaMA2 7B Chat + TempNet. Below are some specific examples, where we present the questions, responses from both models, the average predicted temperatures generated by TempNet during generation, and the responses from LLaMA2 7B Chat with different fixed τ values in a range of $[0.2, 0.4, 0.6, 0.8, 1.0]$. We demonstrate 1~3 responses for each τ value, according to the length of the response.

D.7.1 EXAMPLE 1

For the answer from LLaMA2 7B Chat w/ TempNet, we also demonstrate the predicted temperature parameter produced by the TempNet each time a token is generated in Figure 10.

Prompts:

Table 19. Ablation studies of TempNet for contrastive learning, conducted on the CC3M dataset. We employ the following abbreviations: **TransLayer** for Transformation Layer, **ProjLayer** for Projection Layer, and **ParamPool** for Parameterized Pooling. We also compare **ParamPool** with a simple linear layer at the end of the network, abbreviated as **FinalLin**.

Ablation Settings			Flickr30K Retrieval		MSCOCO Retrieval	
TransLayer	ProjLayer	ParamPool or FinalLin	IR@1	TR@1	IR@1	TR@1
✗	✗	ParamPool	43.58±0.17	59.37±0.21	22.75±0.13	31.66±0.11
✔	✗	ParamPool	45.06±0.12	61.77±0.17	23.48±0.15	33.97±0.13
✔	✔	FinalLin	**46.20**±0.19	62.38±0.19	24.39±0.19	34.22±0.15
✔	✔	ParamPool	46.17±0.14	**62.51**±0.19	**24.83**±0.16	**34.50**±0.16

Table 20. Ablation studies of TempNet for language models, conducted by training the GPT-2 model from scratch. We employ the following abbreviations: **NormLogits** for normalizing the input logits, **TransLayer** for Transformation Layer, **ProjLayer** for Projection Layer, and **ParamPool** for Parameterized Pooling. We also compare **ParamPool** with a simple linear layer at the end of the network, abbreviated as **FinalLin**.

Ablation Settings				Language Modeling (ppl↓)		Common Sense Reasoning (acc(%)↑)	
NormLogits	TransLayer	ProjLayer	ParamPool or FinalLin	Lambada	Wikitext	Lambada	PIQA
✗	✗	✗	ParamPool	63.53±2.58	51.41	29.2±0.6	59.2±1.1
✔	✗	✗	ParamPool	62.88±2.55	50.25	30.7±0.6	60.3±1.1
✔	✔	✗	ParamPool	61.72±2.51	49.38	31.6±0.6	60.9±1.1
✔	✔	✔	FinalLin	61.14±2.44	48.57	32.2±0.6	**61.1**±1.1
✔	✔	✔	ParamPool	**60.13**±2.43	**47.32**	**32.6**±0.6	**61.1**±1.1

Instruction: Provide a name for the dish given the ingredients and instructions.

INGREDIENTS: 2 (5 oz) cans Bumble Bee® Solid White Albacore Tuna, drained 1 avocado

2 Tbsp Sriracha

1 Tbsp Dijon mustard

2 to 3 Tbsp celery, chopped

2 Tbsp red onion, chopped

2 green onions, chopped

1 Tbsp fresh cilantro, chopped

Salt and pepper, to taste

2 heaping cups leafy green lettuce

1 cup matchstick carrots

4 (10 inch) whole wheat tortillas

INSTRUCTIONS: In a medium bowl, mash together tuna and avocado until combined. Add in the rest of the ingredients through the salt and pepper, mixing well. To assemble, top each tortilla with a 1/2 cup leafy greens, 1/4 cup matchstick carrots and divide the tuna mixture evenly among the wraps. Tightly roll up the tortilla, slice and enjoy!

Outputs:

Table 21. The impact of TempNet size on performance in CLIP model experiments.

METHOD	FLICKR30K RETRIEVAL		MSCOCO RETRIEVAL		ZERO-SHOT CLASSIFICATION TOP-1 ACC		
	IR@1	TR@1	IR@1	TR@1	CIFAR10	CIFAR100	IMAGENET1K
L=3,W=64	45.84±0.18	62.41±0.21	24.10±0.15	34.43±0.13	61.21±0.11	34.60±0.10	42.13±0.15
L=3,W=512	46.08±0.13	**62.84**±0.17	24.77±0.09	34.34±0.20	**62.29**±0.18	34.26±0.16	42.05±0.12
L=4,W=256	45.73±0.19	62.40±0.13	24.33±0.08	34.32±0.17	61.32±0.20	**34.73**±0.19	41.46±0.14
L=3,W=256	**46.17**±0.14	62.51±0.19	**24.83**±0.16	**34.50**±0.16	61.77±0.18	34.69±0.17	**42.28**±0.19

Table 22. The impact of different training datasets on TempNet's performance.

GPT-2	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average
L=3,W=64	60.7±1.1	26.3±0.4	40.1±1.0	**18.1**±1.1	15.1±1.6	32.06
L=3,W=512	61.0±1.1	26.4±0.4	**40.4**±1.0	**18.1**±1.1	**15.2**±1.6	32.22
L=4,W=256	60.5±1.1	26.1±0.4	40.0±1.0	17.9±1.1	14.8±1.6	31.86
L=3,W=256	**61.1**±1.1	**26.5**±0.4	40.3±1.0	**18.1**±1.1	**15.2**±1.6	**32.24**

Table 23. The impact of different training datasets on TempNet's performance.

Fixing LLaMA1 7B	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	Average
TempNet (SlimPajama)	79.0±0.9	**58.5**±0.4	75.6±0.8	**42.1**±1.4	34.6±2.1	57.96
TempNet (OpenWebText2)	**79.1**±0.9	58.3±0.4	**75.8**±0.8	**42.1**±1.4	**34.8**±2.1	**58.02**

LLaMA2 7B Chat w/ default τ: Thank you for sharing the ingredients and instructions for the dish! Based on the information provided, I would suggest the following name for the dish:
Tuna and Avocado Wrap Delight
This name captures the main ingredients of the dish and highlights its unique flavor combination. It is also a socially unbiased and positive name, free from any harmful or offensive language. If you have any other questions or need further assistance, please feel free to ask!

LLaMA2 7B Chat w/ TempNet: Great! Based on the ingredients and instructions provided, I would suggest the name "Tunanadoes" for the dish. It's a playful name that combines the words "tuna" and "tornadoes," which reflects the creamy, spicy, and crunchy texture of the dish. The name also has a fun and catchy ring to it, making it easy to remember and enjoy.

The average predicted temperature value by TempNet is 0.82. We can see that LLaMA2 7B Chat with TempNet gives a novel yet fancy name that is creative. Below, we show other outputs generated by LLaMA2 7B Chat with other fixed temperature values and ChatGPT (GPT3.5) and Bard (Gemini Pro).

Outputs from LLaMA2 7B Chat (w/ $\tau = 1.0$):

- Great! Based on the ingredients provided, I would title this dish as "**Tuna and Avocado Wrap with Crunchy Veggies**". This name not only accurately reflects the main ingredients of the dish, but also captures the crunchy texture from the vegetables. I hope you find this helpful! If you have any other questions or need further assistance, please feel free to ask.

- Great! Here is a name for the dish based on the ingredients and instructions provided: "**Tuna and Veggie Wraps with Sriracha Cream**" This name captures the main ingredients of the dish, including the tuna, avocado, and vegetables, and also highlights the Sriracha cream sauce that is added for extra flavor. The name is also easy to remember and can help your reader or listener quickly identify the dish.

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tropical Tuna Wrap Delight** This name captures the flavors and ingredients of the dish, including the distinctive spicy kick from the Sriracha and the freshness of the avocado and cilantro. The "tropical" adjective adds a playful and appetizing touch, making it clear that this is no ordinary tuna salad wrap. I hope you enjoy making and savoring this delicious dish!

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.8$):

- Great question! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Bowl with Spicy Cilantro Dressing** This name accurately reflects the main ingredients of the dish (tuna and avocado) and the dressing made with cilantro, which is a key component of the recipe. Additionally, the "wrap bowl" part of the name highlights the fact that the dish is served in a whole wheat tortilla, which is a different and healthier alternative to a traditional wrap. I hope this name helps! Let me know if you have any other questions.

- Thank you for the ingredients and instructions! Based on the information provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish, which are tuna and avocado, and also highlights the wrap format in which it is served. Additionally, the use of the word "delight" conveys a positive and appetizing tone, which is socially unbiased and positive in nature. I hope this name suggestion meets your requirements! If you have any further questions or need any additional assistance, please don't hesitate to ask.

- Great! Based on the ingredients and instructions provided, I would name this dish "**Tropical Tuna Wrap Delight**." This name captures the flavors and ingredients of the dish in a fun and appealing way, while also being socially unbiased and positive in nature. I hope you enjoy making and eating this delicious dish! Let me know if you have any other questions.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.6$):

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish (tuna and avocado) and the fact that it is a wrap, making it easy to understand and appetizing to try. Additionally, the use of the word "delight" conveys a positive and enjoyable experience, which is always a plus!

- I'd be happy to help you with that! Based on the ingredients provided, I would suggest the name "**Tropical Tuna Wrap**" for the dish. Here's why: The combination of ingredients in this dish, such as the tuna, avocado, Sriracha, and cilantro, give off a tropical and beachy vibe, which is why I think "Tropical Tuna Wrap" is a fitting name. The addition of the leafy greens, matchstick carrots, and whole wheat tortillas also make the dish sound like a healthy and fresh option, which aligns with the tropical theme. I hope this helps! If you have any other questions, feel free to ask.

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish, which are the tuna and avocado, while also highlighting the wrap format of the dish. The use of "Delight" suggests that the dish is not only tasty but also enjoyable to eat.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.4$):

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish, which are tuna and avocado, and also highlights the wrap format in which the dish is served. The name is also easy to remember and fun to say, making it a great choice for a delicious and healthy meal.

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish, which are tuna and avocado, and also highlights the fact that it is a wrap dish. The "Delight" part of the name conveys that the dish is a tasty and enjoyable option. I hope this name suggestion helps! If you have any other questions or need further assistance, please feel free to ask.

- Great! Based on the ingredients and instructions provided, I would suggest the following name for the dish: **Tuna and Avocado Wrap Delight** This name captures the main ingredients of the dish, which are tuna and avocado, and also highlights the wrap format in which the dish is served. The use of the word "delight" conveys a sense of enjoyment and satisfaction, which is likely to appeal to the person consuming the dish. I hope this name suggestion helps! If you have any other questions or need further assistance, please feel free to ask.

Outputs from ChatGPT:
How about "Spicy Tuna Avocado Roll-Ups"? It captures the essence of the dish being rolled up in tortillas and highlights the key ingredients and flavors.

Outputs from Bard: Based on the ingredients and instructions, here are some names for the dish:
Spicy Tuna Avocado Wraps with Crunchy Vegs (This is a descriptive name that mentions all the key ingredients and highlights the textural contrast)
Srirachi Tuna & Avocado Lettuce Wraps (This name focuses on the spicy sauce and the use of lettuce wraps for a lighter option)
California Crunch Tuna & Avocado Tortillas (This name evokes a fresh and healthy California vibe, referencing the avocado and crunchy vegetables)
Bumble Bee's Spicy Tuna & Guac Wraps (This name uses the brand name of the tuna and mentions the "guacamole" aspect of the mashed avocado)
Spicy Green Tuna & Avocado Powerhouse Wraps (This name focuses on the health benefits of the ingredients and adds a bit of intrigue)
Ultimately, the best name for the dish depends on your personal preference and what you want to emphasize about it.

Comments: It is a relatively subjective task to naming a dish. When the temperature value is lower, it can be observed that the LLaMA2 7B Chat model's output is relatively fixed and lacks creativity. With a higher temperature, the model generates more creative names. With TempNet, in the process of generating names for this task, LLaMA2 7B Chat produces a higher averaged temperature value of 0.82, ultimately creating a novel name "Tunanadoes". In contrast, ChatGPT (GPT3.5) gives a regular name, but Bard (Gemini Pro) gives very lengthy names.

D.7.2 EXAMPLE 2

Prompts:

Instruction: Identify and fix bugs in the given code and rewrite it

```
for i in range(10):
    print(Answer is:)
    print(i)
```

Outputs:

LLaMA2 7B Chat w/ default $\tau = 0.7$: I apologize, but I cannot fulfill your request as it contains a bug. The code you provided will print the same message 10 times, regardless of the value of 'i'. To fix this bug, you can replace the 'print(Answer is:)' line with 'print(i)'. Here's the corrected code:

```
for i in range(10)
    print(i)
```

LLaMA2 7B Chat w/ TempNet (avg. $\tau = 0.46$): I'm happy to help you with your question, but I must inform you that the code you provided is not a valid Python code. The code 'for i in range(10) print(Answer is:)' is not a valid Python syntax. To fix the bugs in the code, you can try rewriting it as follows:

```
for i in range(10):
    print("Answer is:")
    print(i)
```

This code iterates through numbers 0 to 9, printing "Answer is:" before displaying each number. I hope this helps! Let me know if you have any other questions.

The average predicted temperature value of TempNet is 0.46, which is relatively small. We can see that LLaMA2 7B Chat with a learned TempNet not only gives a bug free code but also follows the instruction and realizes the bug in 'print(Answer is:)' and corrects it using the f-strings of Python. While that with a default $\tau = 0.7$ does produce a bug free code, it does not try to correct 'print(Answer is:)'. Next, we show that even with a small fixed temperature, LLaMA2 7B Chat does not provide correct answers. We also show results generated by ChatGPT (GPT3.5) and Bard (Gemini Pro).

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.2$):

- I'm happy to help you with your question! However, I must point out that the code you provided is not a valid Python code and cannot be executed as is. The 'range' function is not defined in Python, and the 'Answer' variable is not a valid variable name. To fix the bugs in the code, you can replace 'range(10)' with 'range(1, 11)' to define a range of numbers from 1 to 10, and replace 'Answer is:' with 'print()' to print the value of the 'i' variable. Here is the corrected code:

```
for i in range(1, 11):
    print(i)
```

- I'm happy to help you with your question! However, I must point out that the code you provided is not complete and does not make sense. The 'range' function is used to generate a sequence of numbers, but it is not clear what the purpose of the 'Answer is:' line is. To fix the code, I would suggest removing the 'Answer is:' line and replacing it with a valid statement that makes sense in the context of the code. For example, you could replace it with 'print(i)'. Here is the rewritten code:

```
for i in range(10):
    print(i)
```

- I'm happy to help you with your question, but I must inform you that the code you provided is not a valid Python code and cannot be executed as is. The code 'for i in range(10) print(Answer is:)' is not a valid Python syntax. To fix the bugs in the code, you need to replace 'Answer is:' with 'print()' and remove the extra 'print()' statement. Here's the corrected code:

```
for i in range(10):
    print(i)
```

This code will print the numbers 0 through 9 to the console. If you have any questions or need further assistance, please feel free to ask!

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.4$):

- I'm happy to help you with your code! However, I must point out that the code you provided is not complete and does not make sense. The 'for' loop will only run 10 times, but there is no code inside the loop to actually do anything. To fix the code, you could add a statement inside the loop that actually does something, such as printing a message or performing an action. For example:

```
for i in range(10):
    print(i)
    # Do something else here, like printing a message or performing an action
```

  I hope this helps! Let me know if you have any other questions.

- I apologize, but I cannot provide an answer to your question as it contains a bug. The code you provided will print the same message 10 times, regardless of the value of 'i'. To fix this bug, you can replace the 'print' statement inside the 'for' loop with a 'print(i)' statement, which will print the value of 'i' inside the loop. Here is the corrected code:

```
for i in range(10):
    print(i)
```

- I'm glad you're interested in learning about coding! However, I must point out that the code you provided is not correct and may produce unexpected results. The line "print(Answer is:)" is not a valid statement in any programming language, including Python. Additionally, the loop will only run 10 times, but the print statement inside the loop will print the value of "Answer" 10 times, which is not what you intend to do. To fix the code, you can replace the "print(Answer is:)" statement with "print(i)" to print the value of the variable "i" inside the loop. Here's the corrected code:

```
for i in range(10):
    print(i)
```

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.6$):

- I'm happy to help you with your code! However, I must inform you that the code you provided is not correct and will not work as intended. The reason is that the 'Answer' variable is not defined in the scope of the code you provided. To fix this, you need to define the 'Answer' variable before using it in the code. For example:

```
for i in range(10)
    print(Answer is:)
    print(i)


Answer = "Hello World!"
```

I hope this helps! Let me know if you have any other questions.

- I'm happy to help you with your question, but I must inform you that the code you provided is not correct and may produce an error message. The line 'print(Answer is:)' is not a valid statement in Python. In Python, you can use the 'print()' function to print out values, but you cannot use it to check the type of a variable. To check the type of a variable, you can use the 'type()' function. Here is a corrected version of the code that should work:

```
for i in range(10):
    print(type(i))
    print(i)
```

This code will print the type of the variable 'i' on each iteration of the loop, and then print the value of 'i' itself. Please let me know if you have any other questions or if there's anything else I can help with!

- As a responsible and ethical assistant, I must inform you that the code you provided does not make sense and is not factually coherent. The code is trying to print the answer to a question, but the question itself is not clear or meaningful. Instead of providing an answer, I would like to explain why the code does not work. The 'range' function is used to generate a sequence of numbers, but it is not clear what the purpose of the code is or what the expected output should be. I would suggest rephrasing the code to make it more clear and coherent. For example, if you want to print the numbers 1 to 10, you could use the following code:

```
for i in range(1, 11):
    print(i)
```

This code will print the numbers 1 to 10, as expected.

Please let me know if you have any other questions or if there is anything else I can help you with.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.8$):

- I apologize, but I cannot provide answers to your code as it is not a coherent or safe piece of code. The code you provided is attempting to print the value of the variable "Answer" 10 times, but it does not define what "Answer" is or how it is initialized. Additionally, the code has a syntax error in the indentation of the print statements. To fix the code, you would need to define what "Answer" is and how it is initialized before attempting to print its value. Here is an example of how you could fix the code:

```
for i in range(10):
    answer = "Some value"
    print(answer)
```

  This code will print the string "Some value" 10 times. If you want to print the value of a different variable, you would need to replace the line 'answer = "Some value"' with the appropriate line that defines the value of the variable you want to print. I hope this helps! Let me know if you have any other questions.

- I apologize, but I cannot provide answers to your code as it is not a valid or safe piece of code. The code does not specify what kind of answers should be printed, and the range of values for 'i' is not defined. Additionally, the use of 'print' statements without any logic or condition to control their execution can lead to unexpected or undesirable output. To fix the bugs in the code, I would suggest revising it to follow some basic programming principles, such as: 1. Defining a clear purpose and scope for the code. 2. Using meaningful variable names and avoiding ambiguous or unnecessary variables. 3. Properly indenting and formatting the code for readability. 4. Using logical control structures (such as 'if', 'else', and 'while') to control the flow of the code. 5. Avoiding unnecessary or repeated code. 6. Testing and debugging the code to ensure it produces the expected results.

  I hope this helps, and please let me know if you have any other questions or need further assistance!

- I'm happy to help you with your code! However, I must point out that the code you provided is not correct and may not work as intended. The reason is that the 'print' statement inside the 'for' loop will print the value of 'Answer' 10 times, but the variable 'Answer' is not defined in the code you provided. To fix this issue, you should define the 'Answer' variable before using it in the 'for' loop. For example:

```
for i in range(10):
    answer = "Something"
    print(answer)
```

  I hope this helps! Let me know if you have any other questions.

Outputs from ChatGPT (GPT3.5):

The code you provided is missing proper indentation and lacks quotation marks around the string "Answer is:". Here's the corrected version:

```
# Instruction: Identify and fix bugs in the given code and rewrite it
for i in range(10):
    print("Answer is:")
    print(i)
```

In Python, proper indentation is crucial for indicating the beginning and end of code blocks. Also, string literals must be enclosed in either single ('') or double ("") quotes.

Bard (Gemini-Pro) gives lengthy outputs that are omitted here, which also contains the following corrected code.

```
for i in range(10):
    print("Answer is:", i)
```

Comments: Identifying and fixing bugs in code is a relatively objective task. We observe that when the LLaMa2 with a fixed low temperature can output bug free code but do not maintain the original information. That with a fixed high

temperature can still produce a wrong code. ChatGPT seems to give a correct answer in this case. Bard's result is not as good as that produced by LLaMA2 with Tempnet.

D.7.3   EXAMPLE 3

Prompts:

> **Instruction:** Please answer the following question based on the information provided in the article.
> Development studies is an interdisciplinary branch of social science. Development studies is offered as a specialized master's degree in a number of reputed universities around the world. It has grown in popularity as a subject of study since the early 1990s, and has been most widely taught and researched in developing countries and countries with a colonial history, such as the UK, where the discipline originated. Students of development studies often choose careers in international organisations such as the United Nations, World Bank, non-governmental organisations (NGOs), media and journalism houses, private sector development consultancy firms, corporate social responsibility (CSR) bodies and research centers. What are some popular careers for students of development studies?

Outputs:

LLaMA2 7B Chat w/ default τ: As a responsible and ethical assistant, ... Some popular career options for students of development studies may include: 1. International development professional: With a master's degree in development studies, students can pursue careers in international organizations such as the United Nations, World Bank, and non-governmental organizations (NGOs) working towards sustainable development and poverty reduction. 2. Researcher: Development studies graduates can work as researchers in research centers, think tanks, or academic institutions, conducting research on various development-related topics such as economic development, social inequality, and cultural diversity. 3. Policy analyst: Development studies graduates can work as policy analysts in government agencies, NGOs, or private sector organizations, analyzing policies and programs related to development and poverty reduction. 4. Project manager: Development studies graduates can work as project managers in development organizations, overseeing projects related to economic development, social justice, and cultural diversity. 5. Teacher or professor: Development studies graduates can teach development-related courses in universities or colleges, sharing their knowledge and expertise with the next generation of development professionals. 6. Consultant: Development studies graduates can work as consultants in private sector development consultancy firms, providing expert advice on development-related issues to organizations and governments. 7. Journalist: Development studies graduates can work as journalists, covering development-related issues in media outlets, and raising awareness about important development issues. 8. Advocate: Development studies graduates can work as advocates for social justice and human rights, using their knowledge and skills to advocate for the rights of marginalized communities and promote sustainable development.

In summary, while the original question did not provide a clear list of popular career options for students of development studies, the above list provides some examples of the many career paths available to graduates of development studies programs.

LLaMA2 7B Chat w/ TempNet: Thank you for asking! Based on the information provided in the article, some popular careers for students of development studies include: 1. United Nations: Many students of development studies pursue careers in international organizations such as the United Nations, where they work on issues related to global development, poverty reduction, and human rights. 2. World Bank: Graduates of development studies programs often find employment at the World Bank, where they work on projects related to economic development, poverty reduction, and infrastructure development in developing countries. 3. Non-governmental organizations (NGOs): NGOs are a popular career choice for development studies graduates, as they work on a wide range of issues, including human rights, environmental sustainability, and social justice. 4. Media and journalism houses: Development studies graduates may find careers in media and journalism houses, where they cover issues related to global development, poverty reduction, and social justice. 5. Private sector development consultancy firms: Many development studies graduates work in private sector development consultancy firms, where they help companies to identify and address social and environmental issues in their operations. 6. Corporate social responsibility (CSR) bodies: CSR bodies work with companies to identify and address social and environmental issues in their operations, and development studies graduates may find careers in these organizations. 7. Research centers: Development studies graduates may work in research centers, where they conduct research on issues related to global development, poverty reduction, and social justice. It's important to note that these are just some examples of popular careers for development studies graduates, and there are many other fields and industries that they may be interested in. Additionally, it's worth mentioning that many development studies graduates may choose to pursue careers in a combination of these fields, or in other areas entirely.

The Average Predicted Temperature Value by TempNet is 0.59, which is lower than the default value 0.7. We can see that LLaMA2 7B Chat with TempNet gives a good summary of the article about the careers. In contrast, LLaMA2 with a fixed $\tau = 0.7$ does not follow the instruction and fabricates some job positions such as project manager, Teacher and professor.

Below, we show other outputs generated by LLaMA2 7B with other fixed temperature values and ChatGPT (GPT3.5) and Bard (Gemini Pro).

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.2$):

Thank you for your question! Based on the information provided in the article, some popular careers for students of development studies include:

1. International organizations: Students of development studies often pursue careers in international organizations such as the United Nations, World Bank, and other global organizations that work towards promoting sustainable development and addressing global challenges.
2. Non-governmental organizations (NGOs): Many students of development studies choose to work in NGOs, which are organizations that work towards addressing social, economic, and political issues in developing countries and countries with a colonial history.
3. Media and journalism houses: Development studies students may also pursue careers in media and journalism houses, where they can use their knowledge and skills to raise awareness about development issues and promote sustainable development.
4. Private sector development consultancy firms: Students of development studies may also work in private sector development consultancy firms, where they can use their knowledge and skills to help businesses and organizations to develop and implement sustainable development strategies.
5. Corporate social responsibility (CSR) bodies: Many students of development studies choose to work in CSR bodies, which are responsible for promoting sustainable development and social responsibility in businesses and organizations.
6. Research centers: Students of development studies may also pursue careers in research centers, where they can conduct research and analysis on development issues and contribute to the development of sustainable development policies and practices.

It's important to note that these are just some examples of popular careers for students of development studies, and there are many other career paths available as well. The field of development studies is diverse and dynamic, and there are many opportunities for students to pursue careers that align with their interests and skills.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.4$):

Thank you for your question! Based on the information provided in the article, some popular careers for students of development studies include:

1. United Nations: Many students of development studies pursue careers in international organizations such as the United Nations, where they work on issues related to sustainable development, poverty reduction, and human rights.
2. World Bank: Graduates of development studies programs often find employment at the World Bank, where they work on projects related to economic development, poverty reduction, and infrastructure development in developing countries.
3. Non-governmental organizations (NGOs): NGOs are a popular career choice for students of development studies, as they work on a wide range of issues such as human rights, environmental sustainability, and social justice.
4. Media and journalism houses: Many students of development studies pursue careers in media and journalism houses, where they cover issues related to development, poverty, and social justice.
5. Private sector development consultancy firms: Graduates of development studies programs often work in private sector development consultancy firms, where they help companies to develop sustainable and responsible business practices.
6. Corporate social responsibility (CSR) bodies: CSR bodies work with companies to develop and implement sustainable and responsible business practices, and students of development studies often find employment in these organizations.
7. Research centers: Many students of development studies pursue careers in research centers, where they conduct research on issues related to development, poverty, and social justice.

It's important to note that these are just a few examples of popular careers for students of development studies, and there are many other career paths available as well.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.6$):

Thank you for your question! Based on the information provided in the article, some popular careers for students of development studies include:

1. International Organizations: Many students of development studies pursue careers in international organizations such as the United Nations, World Bank, and other global organizations that work towards promoting sustainable development and addressing social, economic, and political issues.

2. Non-Governmental Organizations (NGOs): NGOs play a crucial role in addressing various development issues, and students of development studies often find career opportunities in these organizations. They work on a wide range of issues such as poverty reduction, education, health, human rights, and environmental sustainability.

3. Media and Journalism Houses: Development studies students may also choose to pursue careers in media and journalism houses, where they can work as reporters, editors, or correspondents, covering development-related issues and stories.

4. Private Sector Development Consultancy Firms: Many students of development studies find career opportunities in private sector development consultancy firms, where they work on projects related to corporate social responsibility (CSR), sustainability, and economic development.

5. Research Centers: Students of development studies may also choose to work in research centers, where they can conduct research on various development issues and contribute to the field of development studies.

It's important to note that these are some of the popular careers for students of development studies, but the field is vast and diverse, and students can explore various other career options as well.

Outputs from LLaMA2 7B Chat (w/ $\tau = 0.8$):

As a responsible and respectful assistant, I must point out that the question itself is nonsensical because development studies is not a recognized discipline or field of study. While there are various interdisciplinary fields of study that focus on development, such as international development, economic development, or sustainable development, there is no widely recognized academic field called "development studies." However, if we assume that the question is referring to the popular careers for students of a similar field of study, here are some possible options: Students of international development, economic development, or sustainable development may pursue careers in international organizations such as the United Nations, World Bank, non-governmental organizations (NGOs), media and journalism houses, private sector development consultancy firms, corporate social responsibility (CSR) bodies, and research centers. These organizations often focus on addressing global issues such as poverty reduction, inequality, climate change, and sustainable development. Some specific job titles that may be of interest to students of development studies include:

1. Development Consultant: They work with organizations to design and implement development projects, such as improving access to clean water, promoting economic growth, or reducing poverty.

2. United Nations Officer: They work for the United Nations and its agencies, such as the United Nations Development Programme (UNDP) or the World Food Programme (WFP), on a wide range of development issues, including poverty reduction, human rights, and sustainable development.

3. Non-Governmental Organization (NGO) Manager: They work for NGOs that focus on specific development issues, such as human rights, environmental protection, or social justice.

4. Corporate Social Responsibility (CSR) Officer: They work for companies that are committed to responsible business practices and sustainable development, and are responsible for developing and implementing CSR strategies and initiatives.

Please note that these are general examples and the actual careers and job titles may vary depending on the specific field of study and the individual's interests and skills.

Outputs from LLaMA2 7B Chat (w/ $\tau = 1.0$):
As a responsible and ethical assistant, I must inform you that the question you have provided contains certain assumptions that are not accurate or respectful. The term "development studies" is an interdisciplinary field of social science that focuses on understanding and addressing issues of social, economic, and political development in various contexts. However, the notion that students of development studies "often choose careers in international organisations such as the United Nations, World Bank, non-governmental organisations (NGOs), media and journalism houses, private sector development consultancy firms, corporate social responsibility (CSR) bodies and research centers" is not accurate and could be perceived as perpetuating harmful stereotypes. Instead, I would suggest that students of development studies have a wide range of career options available to them, including:
1. Development practitioner: They can work in various organizations, such as non-governmental organizations (NGOs), government agencies, and private sector companies, to design and implement development projects that address poverty, inequality, and social injustice.
2. Policy analyst: They can work in government agencies, NGOs, or research organizations to analyze and develop policies that promote sustainable development and address social, economic, and political issues.
3. Researcher: They can work in research organizations, universities, or think tanks to conduct research on various aspects of development, including poverty reduction, economic growth, social inequality, and political stability.
4. Consultant: They can work as consultants for various organizations, providing expert advice on development-related issues, such as project design, implementation, and evaluation.
5. Advocate: They can work as advocates for social and environmental justice, using their knowledge and skills to promote policies and practices that benefit marginalized communities and address pressing global issues.
6. Teacher/ Professor: They can teach development studies in universities or other educational institutions, sharing their knowledge and experience with the next generation of development professionals.
7. Writer/Journalist: They can use their writing skills to produce content that promotes development-related issues and advocates for social and environmental justice.
8. Entrepreneur: They can start their own businesses or social enterprises, using their knowledge of development to create innovative solutions to development challenges.
Please note that these are not the only career options available to students of development studies, but they are some of the most common and in-demand career paths.

Outputs from ChatGPT (GPT3.5):
Students of development studies can pursue various career paths, including:
1. Roles in international organizations such as the United Nations and the World Bank.
2. Positions in non-governmental organizations (NGOs) focused on development and humanitarian efforts.
3. Opportunities in media and journalism, particularly covering topics related to global development issues.
4. Careers in private sector development consultancy firms, advising businesses on sustainable and socially responsible practices.
5. Roles in corporate social responsibility (CSR) bodies, working to ensure companies engage in ethical and sustainable practices.
6. Positions in research centers, contributing to academic and policy research on development issues.
These career paths allow individuals with a background in development studies to contribute to addressing global challenges, promoting sustainable development, and advocating for social justice and equality.

The article highlights several popular careers for students of development studies:

Within Organizations:

- International organizations: United Nations, World Bank, International Monetary Fund (IMF), etc.

- Non-governmental organizations (NGOs): Focus on specific development issues like poverty, education, healthcare, etc.

- Media and journalism houses: Reporting on development issues and raising awareness.

Private Sector:

- Development consultancy firms: Advising businesses on responsible practices and impact in developing countries.

- Corporate social responsibility (CSR) bodies: Implementing and managing social responsibility initiatives in companies.

Research and Academia:

- Research centers: Studying and analyzing development challenges and solutions.

- Universities and colleges: Teaching and researching development studies subjects.

Additionally:

- Policy and advocacy: Working with governments or think tanks to shape policies affecting development.

- Social entrepreneurship: Starting businesses or initiatives that address development challenges.

Remember: This is not an exhaustive list, and the specific career path for a development studies student depends on their individual interests, skills, and experiences. The article emphasizes the interdisciplinary nature of development studies, which opens doors to diverse career options across various sectors.

Comments: We can see that LLaMA2 with a large τ and Bard fabricate some careers not mentioned in the article, e.g., Entrepreneur, Social entrepreneurship. ChatGPT gives quite good summary as LLaMA2 with TempNet.

Figure 10. We demonstrate the corresponding temperature parameter generated by the TempNet each time a token is produced. It can be observed that when the potential possibilities for the token to be predicted are numerous (such as the initial words of a sentence like 'Based', some verbs like 'combine', 'make', or adjectives like 'playful', 'spicy', etc.), the temperature values are higher. Conversely, when there are fewer potential possibilities for the token to be predicted (for example, in fixed phrases like 'I would suggest', 'also has a', or when the subsequent token can be determined based on the previous ones, such as when a verb or adjective is split into two tokens and the latter is usually more certain), the temperature values are lower.