STOCHASTIC BRIDGES AS EFFECTIVE REGULARIZERS FOR PARAMETER-EFFICIENT TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient tuning methods (PETs) have achieved promising results in tuning large pre-trained language models (PLMs). By formalizing frozen PLMs and additional tunable parameters as systems and controls respectively, PETs can be theoretically grounded to optimal control and further viewed as optimizing terminal cost and running cost in the optimal control literature. Despite the elegance of this theoretical grounding, in practice, existing PETs often ignore the running cost and only optimize the terminal cost, i.e., focus on optimizing the loss function of the output state, regardless of the running cost that depends on the intermediate states. Since it is non-trivial to directly model the intermediate states and design a running cost function, we propose to use latent stochastic bridges to regularize the intermediate states and serve as the running cost of PETs. As the first work to propose regularized PETs that use stochastic bridges as the regularizers (running costs) for intermediate states, we show the effectiveness and generality of this regularization across different tasks, PLMs and PETs. In view of the great potential and capacity, we believe more sophisticated regularizers can be designed for PETs and better performance can be achieved in the future.

1 INTRODUCTION

Recent years have witnessed the dramatic growth of pre-trained language models (PLMs) in various fields (Devlin et al., 2019; Dosovitskiy et al., 2021). As the size of PLM continues to increase, the number of parameters has now even reached hundreds of billions (Brown et al., 2020; Smith et al., 2022), making fine-tuning the whole model both computationally impractical and environmentally unfriendly. In view of this, a variety of Parameter-Efficient Tuning methods (PETs) are proposed (Houlsby et al., 2019; Hu et al., 2022; Zaken et al., 2022; Lester et al., 2021). By simply tuning a small number of additional parameters, PETs can be comparable to full-parameter.

Despite the success of PETs, its underlying mechanism remains an open problem. Recently, several works have proposed to interpret PETs from the perspective of optimal control. Yang & Liu (2022) first show that the optimization process in Prefix Tuning (Li & Liang, 2021) (a typical method of PETs) can be considered as the search for optimal control variables in the context of optimal control, i.e., the prefixes can be seen as the control variables that drive the PLM to the desired output. Ding et al. (2022) go one step further and derive the connection between optimal control and different PETs including Prefix Tuning, Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2022) and BitFit (Zaken et al., 2022). The optimization of PETs' parameters can be considered as minimizing a combination of two cost functions in the optimal control literature: (1) *Terminal cost* $\mathcal{L}_T$, which measures the quality of the terminal state, and (2) *Running cost* $\mathcal{L}_R$, which measures the feasibility of the controlled intermediate hidden states and the control variables. Although the terminal cost $\mathcal{L}_T$ can well correspond to the loss function of the model output, the running cost $\mathcal{L}_R$ receives little attention in the previous works and is only vaguely described as *the regularizers on the parameters of PETs (control variables)*, ignoring the dependency of $\mathcal{L}_R$ on the intermediate states.

In this work, we re-emphasize the importance of the running cost $\mathcal{L}_R$ and demonstrate that PETs can benefit from a well-designed $\mathcal{L}_R$ that regularizes the intermediate hidden states. We begin by assuming that in PLMs, the intermediate hidden states for generating different tokens in a sentence have different dynamics (or trajectories), and the dynamics can be approximated with stochastic processes in a latent space. More specifically, we first freeze the PLM and learn a mapping from the

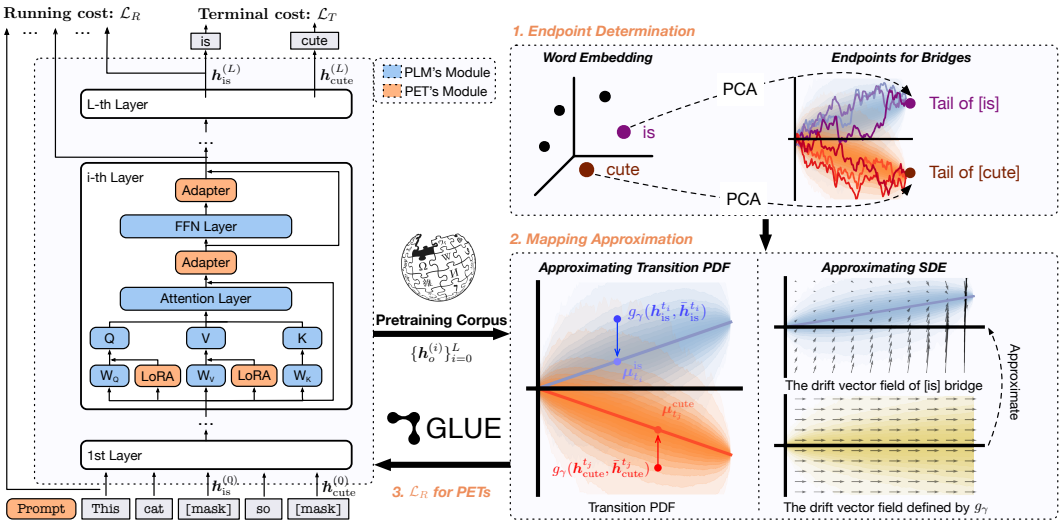


Figure 1: An overview of our proposed latent stochastic bridge regularizer.

original hidden state space to a latent space in which the intermediate hidden states for generating different tokens can be approximated with different *diffusion bridges*. The obtained mapping can then be plugged to the side of the model to regularize the intermediate hidden states when using PETs on downstream tasks. Besides, since a diffusion bridge is a (1) Markov process and (2) a solution to a stochastic differential equation (SDE), we correspondingly propose two methods to learn the latent space: (1) fitting the Markov transition probability density function and (2) fitting the SDE directly. It turns out that the two methods act as a trade-off between efficiency and effectiveness: the first method incurs only negligible computational cost and has satisfactory results, while the second one is slower, but the learned regularizer has greater capability.

We conduct experiments on different PLMs of different sizes, and the experimental results on GLUE (Wang et al., 2019) under both full-set and few-shot settings demonstrate the effectiveness of our proposal across four different PETs. Further analyses show that the regularizer helps pull apart the hidden states of different label words. We also observe that when we project the intermediate hidden states of PETs without our regularizer into our latent space, the better the PETs perform, the closer the latent states is to our latent bridges. The spontaneous approaching behavior indicates there may exists stochastic-bridge-like latent dynamics for well-trained PETs.

In summary, our work has the following contributions: (1) Inspecting PETs from the perspective of optimal control, we re-emphasize the importance of the previously neglected running cost, and propose to use latent stochastic bridges as the regularizer to build the running cost function. (2) We propose two methods to construct the latent space according to the two representations of stochastic bridges, offering the trade-off between efficiency and effectiveness. (3) The latent stochastic process regularizer is shown to be effective across different PLMs, different PETs and different tasks. (4) We show that there may be stochastic-bridge-like latent dynamics for well-trained PETs.

2 Background

2.1 Optimal Control Perspective of PETs

Consider a dataset $\mathcal{D}$ of text-to-text task, given a L-layer PLM with vocabulary $\mathbb{V}$ and a sample $(\boldsymbol{x}, y) \sim \mathcal{D}, \boldsymbol{x} \in \mathbb{V}^N, y \in \mathbb{V}$, where N is the length of $\boldsymbol{x}$.[1] We denote the hidden states of the i-th PLM layer as $\boldsymbol{h}^{(i)} = \{\boldsymbol{h}_j^{(i)}\}_{j=1}^{N}$, where $\boldsymbol{h}_j^{(i)}$ is the state at the position j. Conventionally, adapting PLM to the dataset requires full-parameter fine-tuning, which is given as:

$$\min_{\Delta\theta} \mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}}\Big[\mathcal{L}\big(\boldsymbol{h}^{(L)}, y\big) + \mathcal{R}\big(\Delta\theta\big)\Big], \quad \boldsymbol{h}^{(i)} = \begin{cases} \boldsymbol{h}^{(i-1)} + \mathcal{G}_{\theta+\Delta\theta}^{(i)}\big(\boldsymbol{h}^{(i-1)}\big), & i = 1, 2, \ldots, L, \\ \texttt{Embed}[\boldsymbol{x}], & i = 0, \end{cases} \tag{1}$$

[1]Here we assume $y \in \mathbb{V}$ because a sample where $\boldsymbol{y} \in \mathbb{V}^M$ can be decomposed to M samples with the i-th sample being $([\boldsymbol{x}; \boldsymbol{y}_{<i}], y_i)$ for auto-regressive models, or $([\boldsymbol{x}; \boldsymbol{y}_{-i}], y_i)$ for masked language models.

where θ is the PLM parameter, $\Delta\theta$ is the full-parameter update, $\mathcal{L}(\cdot,\cdot)$ is the loss function, $\mathcal{R}(\cdot)$ is the regularization function, $\mathcal{G}^{(i)}_{\theta+\Delta\theta}(\cdot)$ is the forward propagation of the i-th PLM layer with the parameter $\theta+\Delta\theta$. As the size of θ continues to increase, full-parameter fine-tuning gradually becomes impractical, and different PETs are proposed to mitigate the problem. Let $\phi = \{\phi^{(i)}\}_{i=0}^{L}$ be PETs' parameters. Since $|\phi| \ll |\theta|$, PETs can greatly reduce the tuning cost. More details of specific PETs can be found in Appendix A.

Ding et al. (2022) give a unified view of PETs from the perspective of optimal control, and Equation 1 can be re-written as

$$\min_{\phi} \mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}}\Big[\mathcal{L}_T\big(\boldsymbol{h}^{(L)},y\big)+\sum_{i=0}^{L}\mathcal{L}_R\big(\phi^{(i)}\big)\Big], \quad \boldsymbol{h}^{(i)} = \begin{cases} \boldsymbol{h}^{(i-1)}+\tilde{\mathcal{G}}^{(i)}_{\theta}\big(\boldsymbol{h}^{(i-1)},\phi^{(i)}\big), & i=1,2,\ldots,L, \\ \texttt{Embed}[\boldsymbol{x}], & i=0, \end{cases} \tag{2}$$

where $\tilde{\mathcal{G}}^{(i)}_{\theta}(\cdot,\phi^{(i)})$ represents the forward propagation of the i-th PLM layer intervened by PETs, $\mathcal{L}_T$ is terminal the cost and $\mathcal{L}_R$ is the running cost.

Typically in classification and generation tasks, $\mathcal{L}_T$ corresponds to the cross-entropy loss of the prediction, and $\mathcal{L}_R$ can be seen as a regularizer on PETs' parameters ϕ. However, in the optimal control literature, $\mathcal{L}_R$ depends on not only the control variables ϕ, but also the controlled intermediate hidden states $\{\boldsymbol{h}^{(i)}\}_{i=1}^{L}$. In this paper, we re-emphasize the importance of considering intermediate hidden states to build $\mathcal{L}_R$ (i.e. rewrite $\sum_{i=0}^{L}\mathcal{L}_R\big(\phi^{(i)}\big)$ to $\sum_{i=0}^{L}\mathcal{L}_R\big(\phi^{(i)},\boldsymbol{h}^{(i)}\big)$), and show that exploiting latent stochastic bridges as $\mathcal{L}_R$ improves all PETs we try in Section 4.

2.2 Diffusion Bridges

A diffusion process $X=(X_t)_{t\in[0,T]}$ is a continuous-time Markov process whose sample paths are almost surely continuous. For any $t_x < t_y$, the process is equipped with a transition Probability Density Function (PDF) $p(t_y, y \mid t_x, x)$, which gives the probability density of reaching y at the time t_y given the history of reaching x at the time t_x. A diffusion process is also the solution to an Itô SDE $d\tilde{X}_t = \mu(t,\tilde{X}_t)dt + \sigma(t,\tilde{X}_t)dB_t$, where B_t is a standard Brownian motion, $\mu(\cdot,\cdot)$ is a drift function and $\sigma(\cdot,\cdot)$ is a diffusion function.

A diffusion bridge $X^{T;\beta}$ is a diffusion process conditioning on the path observations of the two endpoints $(0,0)$ and (T,β), i.e., $X_0^{T;\beta}=0$ and $X_T^{T;\beta}=\beta$. By applying Bayes' rule, we obtain the transition PDF for $X^{T;\beta}$ as $p^{T;\beta}(t_y,y\mid t_x,x) = p(t_y,y\mid t_x,x,T,\beta) = p(T,\beta\mid t_y,y)p(t_y,y\mid t_x,x)/p(T,\beta\mid t_x,x)$. In this work, we consider two typical diffusion bridges, the Brownian bridge and the Ornstein-Uhlenbeck bridge (OU bridge). We present here only the properties of the Brownian bridge and leave the properties of OU bridge to Appendix B.

Proposition 2.1 (The properties of the Brownian Bridge)**.** A Brownian bridge $X^{T;\beta}$ with $X_0^{T;\beta}=0$ and $X_T^{T;\beta}=\beta$ is the solution to the following SDE:

$$d\tilde{X}_t = \frac{\beta-\tilde{X}_t}{T-t}dt + dB_t, \quad \tilde{X}_0 = 0. \tag{3}$$

The transition PDF from $X_0^{T;\beta}=0$ to $X_{t_y}^{T;\beta}=y$ is given as

$$p^{T;\beta}(t_y,y\mid 0,0) = \frac{1}{\sqrt{2\pi t(T-t_y)}}\exp\left[-\frac{\Big(y-\frac{t_y}{T}\beta\Big)^2}{2t_y(T-t_y)}\right]. \tag{4}$$

3 Latent Stochastic Bridges Regularizer

3.1 The Overall Framework

Since directly regularizing the intermediate states and constructing the running cost are non-trivial, we introduce a projection from intermediate state space to a latent space, and leverage diffusion bridges as regularizers to calculate running cost. Specifically, given a input sequence $\boldsymbol{x}$ and the position o where the model should output the target y, we define a r-dimensional latent space $\mathbb{U}\subseteq$

$\mathbb{R}^r (r < d)$ as well as a learnable mapping $g_\gamma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{U}$, where g_γ projects the hidden state $\boldsymbol{h}_o^{(i)}$ and its context state $\bar{\boldsymbol{h}}_o^{(i)}$ into the latent space $\mathbb{U}$ at each layer of the PLM. We simply take $\bar{\boldsymbol{h}}_o^{(i)}$ as the average of all hidden states at the i-th layer. We assume that the states in the latent space $\boldsymbol{u} = \{g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}_o^{(i)})\}_{i=0}^L$ form a trajectory that is a sample path from X_y with high probability. X_y is the latent diffusion bridge of the label word y.

Since adapting PLMs to downstream tasks can be seen as transferring knowledge obtained from pre-training to downstream tasks, we argue that the latent dynamics of intermediate states for generating the same token y in pre-training and downstream tasks should be similar. We therefore train the mapping g_γ on pre-training corpus where the backbone PLM was pre-trained, and then apply the mapping to the downstream tasks to encourage the dynamics to be similar to that in pre-training.

Specifically, on the corpus where the PLM is pre-trained, we freeze the PLM and use its hidden states $\boldsymbol{h}_o$ to learn g_γ by maximizing the goodness of approximation for $\boldsymbol{u}$ under the bridge X_y:

$$\gamma \leftarrow \arg\max_{\gamma} \left[\text{goodness-of-approx}\big(\{t_i, g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}^{(i)})\}_{i=0}^L, X_y\big) \right], \tag{5}$$

where goodness-of-approx$(\cdot,\cdot)$ is a function that measures how likely $\{t_i, g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}_o^{(i)})\}_{i=0}^L$ is a sample trajectory of X_y. After learning g_γ, by freezing γ, goodness-of-approx$(\cdot,\cdot)$ is used as the running cost in Eq. 2 for PETs on downstream tasks. Finally, the overall objective for PETs becomes

$$\mathcal{L} = \mathcal{L}_T(\boldsymbol{h}_o^{(L)}, y) + \alpha \cdot \text{goodness-of-approx}\big(\{t_i, g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}^{(i)})\}_{i=0}^L, X_y\big) \tag{6}$$

where the second term is the running cost and α is a hyper-parameter controlling the regularization strength. We include the 0-th layer (input layer), considering that some PETs (e.g., prompt tuning) only act on the 0-th layer. By optimizing Equation 6, PETs need to ensure the final hidden state is capable of predicting y, and keep the latent intermediate hidden states at position o conform to the diffusion bridge X_y. Note that introducing g_γ as the regularizer does not increase the number of trainable parameters for PETs during training since γ is fixed. Also, g_γ does not involve in inference.

The determination of the two endpoints for the diffusion bridges $\{X_y\}_{y\in\mathbb{V}}$ will be introduced in Section 3.2. In Equation 5, different designs for the goodness-of-approx function leads to different learning method, and we will discuss in Section 3.3.

3.2 Determining Endpoints for Diffusion Bridges

An intuitive approach to determine the endpoints for the diffusion bridges for each token is to optimize the endpoints together with the mapping g_γ. However, optimizing endpoints and g_γ jointly admits a trivial solution: endpoints for diffusion bridges of all the tokens are set to be $\mathbf{0} \in \mathbb{R}^r$, and g_γ always outputs $\mathbf{0} \in \mathbb{R}^r$. Since $\mathbf{0}$ is always a point in the sample path of a degenerated diffusion bridge with $\mathbf{0}$ being both ends, the goodness-of-approx can be high, but the diffusion bridges are meaningless. Although sophisticated constraints can be imposed, as the first work using diffusion bridges as regularizers for PETs, we simply pre-determine the endpoints and keep them fixed. We will show the effectiveness of the regularizer even when the endpoints are pre-determined and fixed.

Specifically, we apply PCA to the output word embedding matrix $\boldsymbol{V} \in \mathbb{R}^{|\mathbb{V}|\times d}$ of PLM, obtaining a r-dimensional embedding matrix, and re-normalize each of the embedding to have norm η. Let the resulting embedding matrix be $\tilde{\boldsymbol{V}} \in \mathbb{R}^{|\mathbb{V}|\times r}$. We then use $\tilde{\boldsymbol{V}}$ as the tails of the diffusion bridges, and $\mathbf{0} \in \mathbb{R}^r$ as the heads for all the bridges. The intuition for selecting $\tilde{\boldsymbol{V}}$ as the tails is that the trajectories of the intermediate hidden states for similar words should be close. In $\boldsymbol{V}$, similar tokens are close, and $\tilde{\boldsymbol{V}}$ obtained by PCA can well preserve the token similarity after dimension reduction.

3.3 Fitting the Mapping g_γ

Without loss of generality, we use the Brownian bridges as the diffusion bridges to illustrate the fitting of g_γ. The following statements can be analogous to the OU bridge.

3.3.1 Method 1: Approximating Transition Probability Density Function

From Equation 4, the transition PDF of the Brownian bridge is Gaussian, which indicates

$$X_t^{T;\beta} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2), \text{where} \quad \boldsymbol{\mu}_t = \mathbb{E}\left[X_t^{T;\beta}\right] = \frac{t}{T}\beta, \quad \boldsymbol{\sigma}_t^2 = \text{Var}\left[X_t^{T;\beta}\right] = t(T-t).$$

To make g_γ approximate the transition PDF, we maximize the sum of log-probability of $g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}_o^{(i)})$ under the Gaussian distribution at time t_i for all the layers:

$$\text{Goodness-of-approx} = \sum_{i=0}^{L} \frac{\left[g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}^{(i)}) - \boldsymbol{\mu}_t\right]^2}{2\boldsymbol{\sigma}_t^2} + \text{const}, \quad t_i = \frac{i+1}{L+2}. \tag{7}$$

Here, g_γ can be seen as a mapping from the hidden state space to latent space by predicting the expectation of the Brownian bridge X_y at t_i given the hidden states and context information.

3.3.2 METHOD 2: APPROXIMATING THE SDE

Since the Brownian bridge is the solution to the SDE in Equation 3, we can let g_γ approximate the SDE. However, Brownian bridge is a continuous-time stochastic process, while we only have $L+1$ discrete observations on the path, we thus use natural cubic splines knotted at $\{\boldsymbol{h}_o^{(i)}\}_{i=0}^{L}$ to interpolate $\boldsymbol{h}_o$ at any given time t, and denote it as $\boldsymbol{h}_o^t$. We define the SDE approximated by g_γ as:

$$dZ_t = g_\gamma(\boldsymbol{h}_o^t, \bar{\boldsymbol{h}}^t, t)dt + dB_t. \tag{8}$$

Here, we additionally introduce the dependency on t for g_γ, since time information is shown to be important in previous neural differential equation works (Zhang et al., 2020; Dupont et al., 2019). Following Li et al. (2020); Xu et al. (2022), when two SDEs share the same diffusion function, the KL divergence between the probability measures induced by the two SDEs is finite. This is the case here since the diffusion function $\sigma \equiv 1$ for Equation 8 and Equation 3. The KL divergence between the probability measures μ_Y of Equation 8 and μ_X of Equation 3 can thus be estimated by:

$$D_{\mathrm{KL}}(\mu_X || \mu_Y) = \mathbb{E}_Z\left[\int_0^T \frac{1}{2}\|u(t,\gamma)\|_2^2\right], \text{where}$$

$$u(t,\gamma) = \sigma^{-1}\left(g_\gamma(\boldsymbol{h}_o^t, \bar{\boldsymbol{h}}^t, t) - f^{T;\beta}(t, X_t)\right) = g_\gamma(\boldsymbol{h}_o^t, \bar{\boldsymbol{h}}^t, t) - \frac{\beta - \tilde{X}_t}{T-t}.$$

We can thus use the KL divergence as the goodness-of-approx to optimize the mapping g_γ. Here, g_γ can be seen as a mapping from the hidden state space to latent space by approximating the drift vector field of the underlying Brownian bridge given the hidden states, context and time.

4 EXPERIMENTS

To verify the effectiveness and generality of the latent stochastic bridges regularizer, we conduct experiments on (1) PLMs of different sizes: $\text{BERT}_{\text{large}}$ (340M) (Devlin et al., 2019) and $\text{Deberta}_{\text{xlarge}}$ (750M) (He et al., 2021); (2) different PETs: Prompt tuning, LoRA, BitFit and Adapter; (3) different diffusion bridges: Brownian bridge and OU bridge. We show that the regularizers effectively improve the performance on GLUE (Wang et al., 2019) under both full-set and few-shot setting.

4.1 EXPERIMENTAL SETUPS

Datasets. Since both $\text{BERT}_{\text{large}}$ and $\text{Deberta}_{\text{xlarge}}$ include Wikipedia and BookCorpus (Zhu et al., 2015) in their pre-training corpus, we use these two corpora to train g_γ. We report the average performance and the standard deviation on development set over 3 different runs. We report accuracy for MNLI, SST-2, QNLI, RTE, F1 for MRPC and QQP, and Matthews correlation for CoLA. All the tasks are re-casted to text-to-text format, the PLMs are required to output the label word for each example (e.g., *negative* or *positive* for SST-2). We exclude STS-B since it is a regression task.

Models and PETs. We use the checkpoint release by Megatron (Shoeybi et al., 2019) for $\text{BERT}_{\text{large}}$, and the official v1 checkpoint for $\text{Deberta}_{\text{xlarge}}$. We use a simple three-layer MLP as the g_γ, and we set $T = 1$ in all the experiments. For Prompt tuning, we use a soft prompt of length 20, and append it to the end of each sequence. For LoRA, we apply it to query and value layer in attention. For Adapter, we apply it to the output of attention and feed-forward modules. For BitFit, we tune all the bias terms in linear layers and layer normalization modules. For all PETs, we append a special token `[MASK]` to the end of each sample, and use its output as the prediction. Hereafter, we use **PDF regularizer** to refer to using g_γ fitted by approximating transition PDF, and **SDE regularizer** to refers to using g_γ fitted by approximating the SDE, **vanilla** x to refer to PET x without regularizer.

Table 1: The results on GLUE for $\text{BERT}_{\text{large}}$. The values are the average value of the best performances over three different runs, and the subscripts are the standard deviations. The Δ column shows the difference of the average performance between the vanilla PETs regularized PETs.

PET	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE	Average	Δ
PROMPT	$84.4_{0.1}$	$85.3_{0.3}$	$91.5_{0.1}$	$95.5_{0.1}$	$73.9_{2.4}$	$55.5_{3.4}$	$60.8_{1.5}$	$78.1_{0.6}$	-
+BROWN_PDF	$84.7_{0.2}$	$\mathbf{85.5}_{0.0}$	$\mathbf{91.8}_{0.6}$	$95.7_{0.1}$	$75.4_{0.5}$	$56.4_{3.3}$	$61.5_{2.2}$	$78.7_{0.4}$	0.6
+BROWN_SDE	$\mathbf{84.9}_{0.2}$	$85.4_{0.1}$	$\mathbf{91.8}_{0.4}$	$\mathbf{95.8}_{0.3}$	$\mathbf{78.8}_{1.2}$	$\mathbf{61.}4_{2.9}$	$\mathbf{64.}7_{1.1}$	$\mathbf{80.}4_{0.2}$	**2.3**
LORA	$88.8_{0.1}$	$89.2_{0.2}$	$93.5_{0.2}$	$95.5_{0.1}$	$84.6_{0.4}$	$62.8_{1.6}$	$78.9_{1.6}$	$84.8_{0.3}$	-
+BROWN_PDF	$\mathbf{88.9}_{0.1}$	$\mathbf{89.6}_{0.1}$	$\mathbf{93.9}_{0.1}$	$95.6_{0.2}$	$85.1_{0.7}$	$63.7_{0.5}$	$80.0_{0.5}$	$85.2_{0.1}$	0.4
+BROWN_SDE	$\mathbf{88.9}_{0.1}$	$89.5_{0.1}$	$93.7_{0.1}$	$\mathbf{95.7}_{0.1}$	$\mathbf{86.5}_{1.2}$	$\mathbf{63.9}_{0.4}$	$\mathbf{80.9}_{0.8}$	$\mathbf{85.6}_{0.1}$	**0.8**
BITFIT	$\mathbf{87.9}_{0.2}$	$87.6_{0.1}$	$92.7_{0.2}$	$95.6_{0.1}$	$79.4_{2.3}$	$60.2_{0.8}$	$77.0_{1.5}$	$82.9_{0.3}$	-
+BROWN_PDF	$\mathbf{87.9}_{0.1}$	$\mathbf{87.8}_{0.0}$	$\mathbf{93.0}_{0.2}$	$\mathbf{95.7}_{0.1}$	$83.1_{0.8}$	$60.3_{0.6}$	$\mathbf{78.3}_{0.9}$	$83.7_{0.2}$	0.8
+BROWN_SDE	$\mathbf{87.9}_{0.2}$	$87.7_{0.0}$	$92.8_{0.1}$	$\mathbf{95.7}_{0.1}$	$\mathbf{83.3}_{0.8}$	$\mathbf{61.1}_{1.2}$	$77.7_{1.5}$	$\mathbf{83.8}_{0.3}$	**0.9**
ADAPTER	$88.8_{0.1}$	$89.6_{0.3}$	$93.7_{0.1}$	$95.6_{0.1}$	$83.6_{0.1}$	$60.4_{1.2}$	$79.5_{1.2}$	$84.5_{0.3}$	-
+BROWN_PDF	$\mathbf{89.0}_{0.1}$	$89.7_{0.2}$	$93.8_{0.2}$	$\mathbf{95.8}_{0.1}$	$\mathbf{86.5}_{1.1}$	$\mathbf{62.6}_{0.7}$	$\mathbf{83.2}_{0.2}$	$\mathbf{85.8}_{0.2}$	**1.3**
+BROWN_SDE	$88.9_{0.1}$	$\mathbf{89.8}_{0.1}$	$\mathbf{93.9}_{0.2}$	$\mathbf{95.8}_{0.1}$	$85.9_{0.4}$	$62.3_{1.8}$	$82.2_{0.2}$	$85.5_{0.2}$	1.0

Few-shot Experiments. We randomly sample $2 \times k$ examples from the original training set $\mathcal{D}_{\text{train}}$ for each class. The sampling is performed 5 times with different seeds to form 5 training sets $\{\tilde{\mathcal{D}}^{(i)}_{\text{train}}\}_{i=1}^{5}$ and development sets $\{\tilde{\mathcal{D}}^{(i)}_{\text{dev}}\}_{i=1}^{5}$ with each being k-shot. Each time we train on one of the training set $\tilde{\mathcal{D}}^{(i)}_{\text{train}}$, select the best model on $\tilde{\mathcal{D}}^{(i)}_{\text{dev}}$, and report its performance on original development set $\mathcal{D}_{\text{dev}}$.

Hyper-parameters. We list hyper-parameters in Appendix E. We mainly focus on the difference in performance between vanilla PETs and regularized PETs. Therefore we directly set the hyper-parameters to reasonable values and do not perform much hyper-parameter search. But we ensure the hyper-parameters for vanilla PETs and regularized PETs are the same for a fair comparison.

4.2 FULL-SET RESULTS

The experimental results for $\text{BERT}_{\text{large}}$ are reported in Table 1. Due to space limitation, see Table 7 for the complete results including OU bridge regularizers. The first line of each block in the table is the performance of vanilla PETs, and the rest of the lines are the performances of the regularized PETs. We also conduct the same experiments for $\text{Deberta}_{\text{xlarge}}$ and place the results in Appendix C.

In general, both Brownian and OU bridge, and both PDF and SDE regularizers are able to effectively improve the performance of PETs, showing the effectiveness of our proposed regularizer. In particular, for Prompt tuning, the SDE regularizer with both diffusion bridges yield an increase on the average performance of more than 2 points. We assume that it is because Prompt tuning has far less trainable parameters than other PETs, and it only acts at the input layer, which is far from the supervision signals from the terminal cost $\mathcal{L}_T$. Therefore when provided with the regularization on the hidden states, the prompts receive more guidance and eventually reaching a better local optimal.

Overall, the two diffusion bridges in our experiments do not show much difference. As for the 2 fitting methods, SDE regularizer is generally more effective, especially for Prompt tuning where the number of trainable parameters is restricted. However, we also observe that SDE regularizer is about 3 times slower than PDF regularizer, which brings the trade-off between performance and efficiency. One can expect a better performance by leveraging more sophisticated underlying stochastic bridges, exploring more reasonable endpoints for bridges and designing better mapping g_γ. As the first work using latent stochastic bridges as regularizers, we consider the most straightforward cases and aim to show the great potential of the approach and the importance of regularizing hidden states.

4.3 FEW-SHOT RESULTS

We observe that in Table 1, the improvements brought by our regularizer are more substantial on small datasets MRPC, CoLA and RTE. We assume that this is because in rich-resource datasets, the abundant data itself has provided enough information for the model to learn high quality PETs,

Table 2: The results on GLUE for $\text{BERT}_{\text{large}}$ under 16-shot setting. We exclude CoLA because all PETs fail to give reasonable result in few-shot setting.

PET	MNLI	QQP	QNLI	SST-2	MRPC	RTE	Average	Δ
PROMPT	$38.1_{1.5}$	$53.0_{3.1}$	$51.6_{1.4}$	$70.1_{4.9}$	$50.1_{3.0}$	$48.0_{1.3}$	$51.8_{0.9}$	-
+BROWN_PDF	$38.7_{2.3}$	$54.9_{2.8}$	$52.1_{1.1}$	$75.0_{11.0}$	$\mathbf{52.8}_{2.2}$	$50.8_{3.3}$	$54.0_{1.8}$	2.2
+BROWN_SDE	$\mathbf{40.6}_{0.8}$	$55.4_{2.1}$	$\mathbf{52.9}_{1.6}$	$\mathbf{80.0}_{10.9}$	$51.9_{3.6}$	$\mathbf{51.7}_{3.1}$	$\mathbf{55.4}_{1.5}$	**3.6**
LORA	$48.7_{4.5}$	$59.9_{5.5}$	$53.2_{1.2}$	$90.2_{1.1}$	$53.6_{3.4}$	$64.2_{0.9}$	$61.6_{0.6}$	-
+BROWN_PDF	$52.0_{1.3}$	$62.7_{2.2}$	$55.1_{3.8}$	$\mathbf{91.3}_{0.1}$	$57.4_{5.0}$	$65.3_{1.5}$	$64.0_{0.7}$	2.4
+BROWN_SDE	$\mathbf{54.1}_{0.9}$	$\mathbf{65.5}_{1.4}$	$\mathbf{64.3}_{5.1}$	$91.2_{0.3}$	$\mathbf{60.2}_{3.1}$	$\mathbf{65.8}_{1.4}$	$\mathbf{66.8}_{0.6}$	**5.2**
BITFIT	$48.4_{1.6}$	$56.0_{6.1}$	$51.7_{2.5}$	$90.8_{0.8}$	$52.0_{2.3}$	$61.7_{1.4}$	$60.1_{1.1}$	-
+BROWN_PDF	$48.5_{1.9}$	$56.0_{6.0}$	$53.5_{2.0}$	$\mathbf{91.0}_{0.4}$	$53.9_{2.3}$	$63.2_{1.6}$	$61.0_{0.8}$	0.9
+BROWN_SDE	$\mathbf{52.3}_{0.5}$	$\mathbf{61.2}_{2.9}$	$\mathbf{58.8}_{4.7}$	$90.8_{0.4}$	$\mathbf{54.8}_{3.0}$	$\mathbf{63.9}_{2.5}$	$\mathbf{63.6}_{0.8}$	**3.5**
ADAPTER	$47.4_{3.7}$	$57.0_{7.2}$	$55.8_{2.9}$	$91.0_{0.4}$	$55.8_{2.5}$	$62.7_{2.0}$	$61.6_{1.2}$	-
+BROWN_PDF	$49.0_{4.8}$	$58.5_{7.4}$	$56.9_{3.1}$	$91.4_{0.2}$	$57.2_{4.9}$	$63.2_{3.0}$	$62.7_{1.5}$	1.1
+BROWN_SDE	$\mathbf{52.3}_{2.2}$	$\mathbf{62.4}_{2.9}$	$\mathbf{64.8}_{4.5}$	$\mathbf{91.9}_{0.4}$	$\mathbf{57.3}_{4.1}$	$\mathbf{63.8}_{1.8}$	$\mathbf{65.4}_{1.3}$	**3.8**

while in low-resource datasets, the data is insufficient and the regularizer offers helpful additional supervision. To validate this, we conduct the experiments under the few-shot setting on GLUE.

The 16-shot results are shown in Table 2, and the results for OU bridge, results for 4-, 8- and 32-shot and results for $\text{Deberta}_{\text{xlarge}}$ are placed in Appendix C. On all the PETs, the SDE regularizer yields an improvement of more than 3 points. Particularly, SDE regularizer on LoRA brings an improvement of 5.2. By applying the regularizer in the few-shot setting, there is now a substantial boost on what was originally a rich-resource dataset, such as MNLI, QQP and QNLI. The PDF regularizer also gives modest improvements. Although slightly inferior to the SDE regularizer, it is still satisfying considering that PDF regularizer brings such a performance improvement with little computational cost introduced. We additionally observe that the improvement is more significant on $\text{Deberta}_{\text{xlarge}}$ in Table 9, demonstrating the potential of our regularizer on larger models.

5 ANALYSES

In order to have a better understanding of the role played by our regularizers, we analyze the hidden states of the PETs with and without the regularizer in this section. We choose Prompt tuning as a representative in the analyses. By varying the hyper-parameter α in Equation 6, we show that as the strength of the regularization gets stronger, the clusters of hidden states corresponding to different label word becomes compacter and more distinguishable. Also, we show that the hidden states of vanilla PETs spontaneously approach the latent bridges in latent space without knowing the bridges, which may indicate that there exists intrinsically diffusion-bridge-like latent dynamics for PETs.

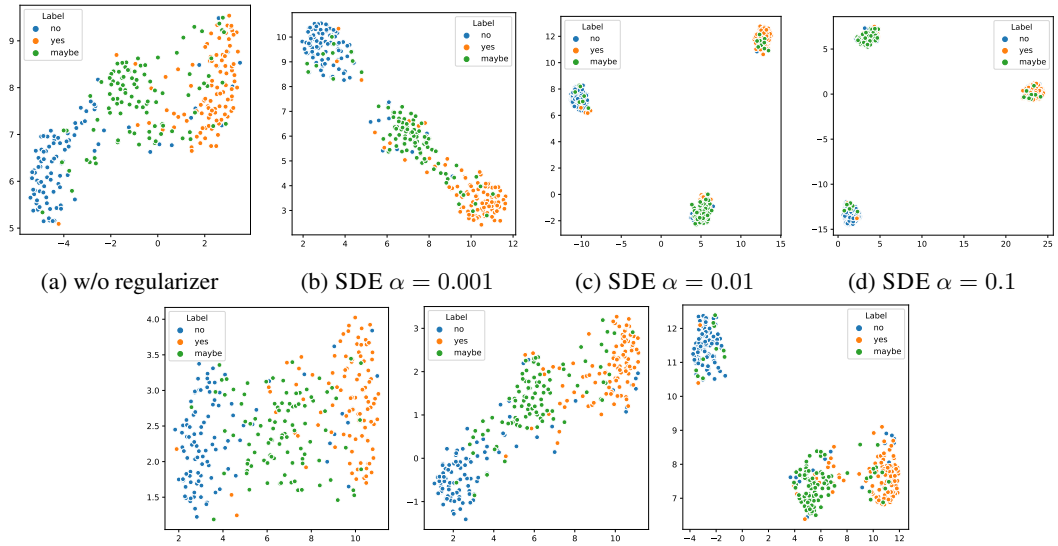
5.1 THE REGULARIZER WIDENS THE DISTANCES BETWEEN LABEL CLUSTERS

We use the different prompts obtained with or without the regularizer on full-set GLUE, and record the intermediate hidden states at the position of prediction as $\{\boldsymbol{h}^{(i)}_{[\text{MASK}]}\}_{i=1}^{L}$. We vary the regularization strength by adjusting the coefficient α in Equation 6 to inspect the impact of regularization strength on the hidden states. Note that when $\alpha = 0$, it degenerates to the vanilla PET.

We randomly sample 100 samples for each label in MNLI, and use UMAP (McInnes et al., 2018) to reduce the dimension of the last layer's hidden states of different prompts, and plot them in Figure 2. It shows clearly that for both regularizers, as the strength of the regularization becomes stronger, the hidden states of the last layer becomes more distinguishable among different labels. By looking at the axes of these plots, we find that the distance between the clusters generally increases when

Table 3: Pearson's correlation between the regularization strength α and the average distance between the centroids of different label clusters. ***: $p < .001$, **: $p < .01$, *: $p < .05$.

	Coef.	
Dataset	PDF	SDE
MNLI	0.928^{***}	0.894^{***}
QQP	0.715^{*}	0.897^{***}
QNLI	0.971^{***}	0.837^{**}
SST-2	0.966^{***}	0.867^{***}
MRPC	0.226	0.433
CoLA	0.633^{*}	0.807^{**}
RTE	0.440	0.589^{*}



(a) w/o regularizer (b) SDE $\alpha = 0.001$ (c) SDE $\alpha = 0.01$ (d) SDE $\alpha = 0.1$

(e) PDF $\alpha = 0.01$ (f) PDF $\alpha = 0.1$ (g) PDF $\alpha = 1.0$

Figure 2: A visualization of the last layer's hidden states on MNLI using the prompt that is trained (a) without regularizer (b-d) with the SDE regularizer (e-g) with the PDF regularizer.

the regularization strength is increased. We also notice that SDE regularizer better helps separate the last layer's hidden states by substantially enlarging the distance between the centroids of different labels, which could be one of the reasons why SDE regularizer has a better effectiveness in almost all our experiments.

We also calculate the Pearson's correlation between the regularization strength α and the average distance between the centroids of different clusters. The results are shown in Table 3. On all the datasets, the regularization strength α has a positive correlation to the average centroid distance, and on most of the datasets, the correlations are significant in the sense that the p-value $< .05$. This indicates that as the regularization strength becomes stronger, the centroids of different label clusters become more distant, which is a desired effect because the regularizer encourages the hidden states for different label words to conform to different latent diffusion bridges.

5.2 Hidden States Approach Spontaneously to the Latent Bridges

An interesting phenomenon we observe is that the vanilla PETs' intermediate hidden states spontaneously approach our latent bridges when they are projected by our mapping g_γ. That is, applying our mapping g_γ to the hidden states of vanilla PETs we find that when the performance of vanilla PETs becomes better, the average distance from $\{g_\gamma(\{\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}^{(i)}\})\}$ to our latent bridge gets closer. Here, similar to Wang et al. (2022), we define the distance from $g_\gamma(\boldsymbol{h}_o^{(i)}, \bar{\boldsymbol{h}}^{(i)})$ to its corresponding latent bridge X_y using Equation 7 without the constant. Note that the vanilla PETs have no access to the g_γ and the latent bridges during the training, and g_γ also has no access to the PETs during its fitting process.

Table 4: Kendall's rank correlation between the number of shots and the distance to the latent bridges.

Dataset	Coef.	p-value
MNLI	-0.39	.026*
QQP	-0.36	.037*
QNLI	-0.32	.069
SST-2	-0.50	.005**
MRPC	-0.21	.225
RTE	-0.34	.049*

We demonstrate the above observation by conducting analyses in few-shot scenarios and reporting the correlation between (1) the number of shots and the average distance from latent hidden states to latent bridges in Table 4 (2) the performance and the average distance from latent hidden states to latent bridges in Table 5. Specifically, we report Kendall's rank correlation[2] for (1), and Pearson's correlation for (2). See Appendix D for the detailed setup for the calculation of the correlations.

[2]The reason that we choose Kendall's rank correlation is that it is suitable for data with ties. See Appendix D.

From Table 4, the number of shots has a negative correlation to the distance, and the correlation is significant on 4 out of 6 datasets. This indicates that as the amount of available data increases for vanilla PETs, its intermediate hidden states in latent space spontaneously approach latent bridges even without knowing the mapping g_γ and the bridges. Additionally, the results in Table 5 show the negative correlation between the performance of vanilla PETs and the distance to the latent bridges, and it is significant on 3 out of 6 datasets.

Table 5: Pearson's correlation between the performance and the distance to the latent bridges.

Dataset	Coef.	p-value
MNLI	-0.09	.715
QQP	-0.42	.062
QNLI	-0.77	$<.001^{***}$
SST-2	-0.69	$<.001^{***}$
MRPC	-0.13	.591
RTE	-0.87	$<.001^{***}$

Altogether, the two findings on correlation show that as the PETs learn to solve the problem, their intermediate hidden states spontaneously approach our latent bridge in the latent space defined by g_γ. This implies that there exists intrinsically diffusion-bridge-like latent dynamics for PETs, and also justifies our use of latent diffusion bridges as regularizers.

6 Related Works

Recent years have witnessed the success of PLMs (Raffel et al., 2020; Brown et al., 2020), which acquires rich knowledge from unlabeled data in a self-supervised manner and can be adapted to specific tasks by tuning their parameters. Despite the success of PLMs, as the size of PLMs continues to grow, it becomes increasingly impractical to perform full-parameter fine-tuning for downstream tasks. Therefore, various recent efforts have been devoted to PETs, aiming to freeze PLMs and only tune a few additional parameters for task adaptation, such as Prompt tuning (Lester et al., 2021) that prepends tunable tokens per task to the input text, Adapter (Houlsby et al., 2019) that inserts small modules into each layer, BitFit (Zaken et al., 2022) that tunes only the bias terms, and LoRA (Hu et al., 2022) that decomposes the weight updates into low-rank matrices. In this paper, based on the theoretical grounding of PETs on optimal control (Yang & Liu, 2022; Ding et al., 2022), we develop stochastic bridges as the regularizer for intermediate hidden states and introduce regularized PETs, showing effectiveness and generality on different PLMs and tasks.

Since we adopt latent stochastic bridges as the regularizer, our work closely relates to continuous-time neural ODEs (Chen et al., 2018; Rubanova et al., 2019) and neural SDEs (Li et al., 2019; Kidger et al., 2021). Continuous-time neural ODEs and SDEs model the dynamics of the hidden states with ODEs and SDEs parameterized by neural network respectively For example Chen et al. (2018) show the resemblance between ODE and ResNet (He et al., 2016), and propose to learn the neural ODEs as the generalization of ResNet in continuous time. Inspired by these works, we use SDEs to represent the latent dynamics of PETs in the latent space. Our work differs from these work in that we focus on using neural SDEs as regularizers for intermediate hidden states, rather than feature extractors on downstream tasks.

We also notice that Wang et al. (2022) explore the use of Brownian bridge in PLMs. However, they use Brownian bridge to regularize the text dynamics across time, while we use Brownian bridge to regularize the dynamics of intermediate hidden states across model's layers. We additionally show that diffusion bridges other than Brownian bridge can be easily applied in our regularizer. As far as we know, our work is the first to show the diffusion-bridge-like dynamics for hidden states across layers and use diffusion bridges as the regularizer for intermediate hidden states.

7 Conclusion

In this work, we start from the optimal control perspective of PETs and we notice that the existing PETs lack a running cost that regularizes the intermediate hidden states. Therefore, we propose to use stochastic bridges in a latent space as the regularizers for PETs. Experimental results on different models, different tasks and different PETs show that the proposed regularizer effectively improve the PETs' performance. Our analyses further show that when the PETs are trained without the regularizer, the hidden states spontaneously approach our diffusion bridges, indicating that there exists intrinsically diffusion-bridge-like dynamics for PETs. As the first work using stochastic bridges as regularizers, we show its effectiveness and generality even with simple diffusion bridges. We believe it is a promising direction and we are excited to see more future work.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6572–6583, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *CoRR*, abs/2203.06904, 2022. doi: 10.48550/arXiv.2203.06904. URL https://doi.org/10.48550/arXiv.2203.06904.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3134–3144, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/21be9a4bd4f81549a9d1d241981cec3c-Abstract.html.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3816–3830. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.295. URL https://doi.org/10.18653/v1/2021.acl-long.295.

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

2016, Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Patrick Kidger, James Foster, Xuechen Li, and Terry J. Lyons. Neural sdes as infinite-dimensional gans. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5453–5463. PMLR, 2021. URL http://proceedings.mlr.press/v139/kidger21b.html.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://doi.org/10.18653/v1/2021.emnlp-main.243.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://doi.org/10.18653/v1/2021.acl-long.353.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients and variational inference for stochastic differential equations. In Cheng Zhang, Francisco J. R. Ruiz, Thang D. Bui, Adji Bousso Dieng, and Dawen Liang (eds.), *Symposium on Advances in Approximate Bayesian Inference, AABI 2019, Vancouver, BC, Canada, December 8, 2019*, volume 118 of *Proceedings of Machine Learning Research*, pp. 1–28. PMLR, 2019. URL http://proceedings.mlr.press/v118/li20a.html.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In Silvia Chiappa and Roberto Calandra (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3870–3882. PMLR, 2020. URL http://proceedings.mlr.press/v108/li20i.html.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL https://doi.org/10.21105/joss.00861.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Yulia Rubanova, Tian Qi Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5321–5331, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/42a6845a557bef704ad8ac9cb4461d43-Abstract.html.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019. URL http://arxiv.org/abs/1909.08053.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990, 2022. URL https://arxiv.org/abs/2201.11990.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Rose E. Wang, Esin Durmus, Noah D. Goodman, and Tatsunori Hashimoto. Language modeling via stochastic processes. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=pMQwKL1yctf.

Winnie Xu, Ricky T. Q. Chen, Xuechen Li, and David Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pp. 721–738. PMLR, 2022. URL https://proceedings.mlr.press/v151/xu22a.html.

Zonghan Yang and Yang Liu. On robust prefix-tuning for text classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=eBCmOocUejf.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 1–9. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-short.1. URL https://doi.org/10.18653/v1/2022.acl-short.1.

Han Zhang, Xi Gao, Jacob Unterman, and Tom Arodz. Approximation capabilities of neural odes and invertible residual networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11086–11095. PMLR, 2020. URL http://proceedings.mlr.press/v119/zhang20h.html.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 19–27. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.11. URL https://doi.org/10.1109/ICCV.2015.11.

A BACKGROUND FOR PARAMETER-EFFICIENT TUNING METHODS

The large number of parameters in the PLMs makes fine-tuning impractical, therefore different PETs are proposed to mitigate the problem. The current PETs can be categorized into three groups: addition-based, specification-based and reparameterization-based (Ding et al., 2022). To verify the generality of our method, we include one or two PETs from each category in this work, and we give a brief review to these PETs.

Prompt Tuning is an addition-based PET. It prepends or appends trainable virtual tokens $\boldsymbol{P} \in \mathbb{R}^{m\times d}$ to each sequence $\boldsymbol{x} \in \mathbb{R}^{n\times d}$ to form a new input sequence $[\boldsymbol{P};\boldsymbol{x}]$ or $[\boldsymbol{x};\boldsymbol{P}] \in \mathbb{R}^{(n+m)\times d}$, where n, m are length of original sequence and virtual tokens respectively, d is the embedding dimension. The virtual tokens $\boldsymbol{P}$ can be either continuous Lester et al. (2021) or be restricted to be embeddings of discrete tokens in vocabulary (Gao et al., 2021).

Adapter is an addition-based PET. It inserts two-layer MLPs after the attention module and feed-forward module at each layer. Denote $\boldsymbol{h} \in \mathbb{R}^d$ as the input of Adapter, r as the intermediate dimension of Adapter's MLP, $\boldsymbol{W}_d \in \mathbb{R}^{r\times d}, \boldsymbol{W}_u \in \mathbb{R}^{d\times r}$ as the down-projection and up-projection of Adapter, and σ as the activation function. Then the computation of Adapter can be formulated as

$$\boldsymbol{h} \leftarrow \boldsymbol{W}_u\sigma(\boldsymbol{W}_d\boldsymbol{x}) + \boldsymbol{h}$$

BitFit is a specification-based PET. It specifies the bias terms in layer normalization modules and linear transformation modules as trainable.

LoRA is a reparameterization-based PET. It assumes that when training the model, the updates $\Delta\boldsymbol{W}$ for model's pre-trained parameters $\boldsymbol{W} \in \mathbb{R}^{d\times k}$ are low-rank, and thus reparameterize the $\Delta\boldsymbol{W}$ of each matrix in attention module with a low-rank decomposition $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{B} \in \mathbb{R}^{d\times r}$, $\boldsymbol{A} \in \mathbb{R}^{r\times k}$. For a forward pass $\boldsymbol{h} = \boldsymbol{W}\boldsymbol{x}$, the computation of LoRA can be written as

$$\boldsymbol{h} = (\boldsymbol{W} + \Delta\boldsymbol{W})\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}$$

B PROPERTIES FOR ORNSTEIN-UHLENBECK BRIDGE

Proposition B.1 (Properties of Ornstein-Uhlenbeck Bridge)**.** A Ornstein-Uhlenbeck $X^{T;\beta}$ pinned at $X_0^{T;\beta} = 0$ and $X_T^{T;\beta} = \beta$ is the solution to the following SDE:

$$d\tilde{X}_t = q\left[-\coth\left[q(T-t)\right]\tilde{X}_t + \frac{\beta}{\sinh\left[q(T-t)\right]}\right]dt + \sigma dB_t, \quad \tilde{X}_0 = 0, \tag{9}$$

where q is the diffusion coefficient and σ is the diffusion for the OU process. The transition probability density function reads as:

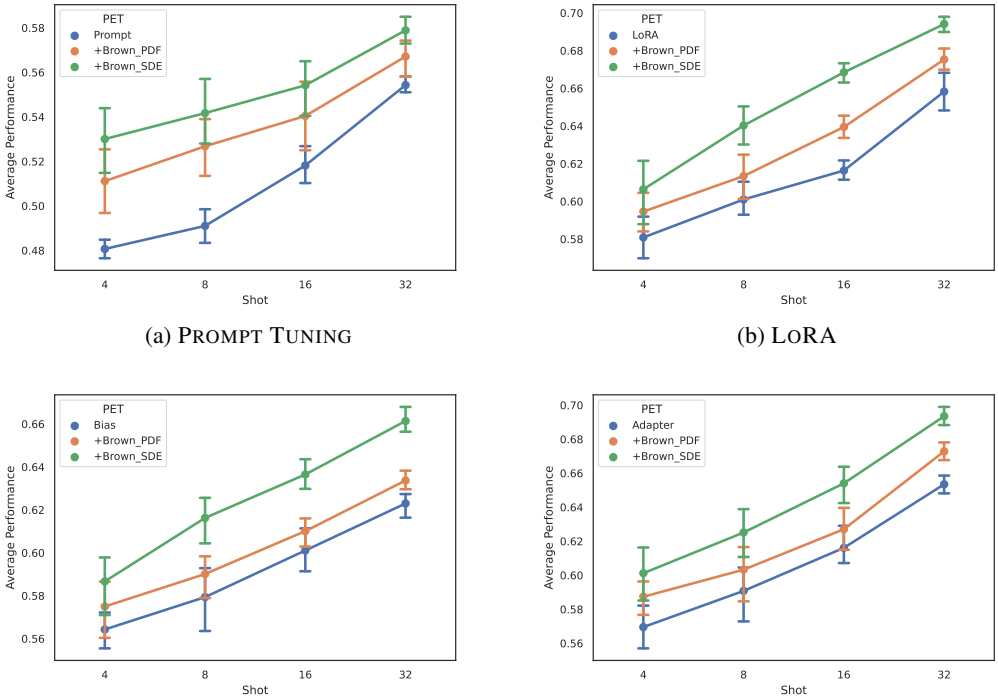
$$p^{T;\beta}(t, y \mid 0, 0) = \frac{1}{\sqrt{2\pi\sigma(s,t)}}\ \exp\left\{-\frac{\left(y - \frac{\sinh(q(T-t))}{\sinh(q(T-s))}\beta\right)^2}{2\sigma(s,t)}\right\}, \tag{10}$$

where

$$\sigma(s,t) = \frac{\sigma^2}{q}\ \frac{\sinh(q(T-t))\sinh(q(t-s))}{\sinh(q(T-s))}. \tag{11}$$

C OTHER RESULTS FOR GLUE EXPERIMENTS

In this section, we present the complete results including OU bridge regularizer for Table 1 and Table 2. We also report the results for $\text{Deberta}_{\text{xlarge}}$, and the results on few-shot GLUE for both $\text{BERT}_{\text{large}}$ and $\text{Deberta}_{\text{xlarge}}$ under 4-, 8-, 16-, and 32-shot. We observe that $\text{BERT}_{\text{large}}$ cannot give reasonable answers on CoLA and the Matthews correlations are around 0 for all the PETs and all the shots we have experienced with. However, the situation gets better for the larger model $\text{Deberta}_{\text{xlarge}}$. Therefore, we exclude CoLA for $\text{BERT}_{\text{large}}$ and keep it for $\text{Deberta}_{\text{xlarge}}$. We only select the Brownian bridge as the representative in this section, since the Brownian bridge and Ornstein-Uhlenbeck bridge have no significant difference in Table 1.



(a) PROMPT TUNING (b) LORA

(c) BIAS TUNING (d) ADAPTER

Figure 3: The average $\text{BERT}_{\text{large}}$ few-shot GLUE results trained with different PETs under different shots. The results are averaged across 5 different seeds and the error bars indicate the 95% confidence. SDE regularizer consistently outperforms the baseline PDF regularizer.

In Table 7 and Table 6, we report the performance of OU bridge regularizers. The experimental setups are the same as Table 1 and Table 2 respectively. The performances between OU bridge and Brownian bridge do not have a significant difference. In Table 8, we report the performance of $\text{Deberta}_{\text{xlarge}}$ on full GLUE datasets. On all four PETs, the SDE regularizer outperforms the PDF regularizer, this is consistent with the results we see in Table 1. The results for 4-, 8-, and 32-shot for $\text{BERT}_{\text{large}}$ and $\text{Deberta}_{\text{xlarge}}$ are plotted respectively in Figure 3 and Figure 4. For simplicity, we only plot the average performance for each PET. We report the results for 16-shot experiments for $\text{Deberta}_{\text{xlarge}}$ in Table 9. The setup for the experiment is almost the same as the experiment in Section 4.3, and the hyper-parameters are listed in Appendix E. The SDE regularizer outperforms the PDF regularizer on most of the PETs except Prompt tuning. We notice that the SDE regularizer helps $\text{Deberta}_{\text{xlarge}}$ substantially on CoLA for most of the PETs, indicating the SDE regularizer can effectively provide useful guidance when the data is scarce and the task is hard.

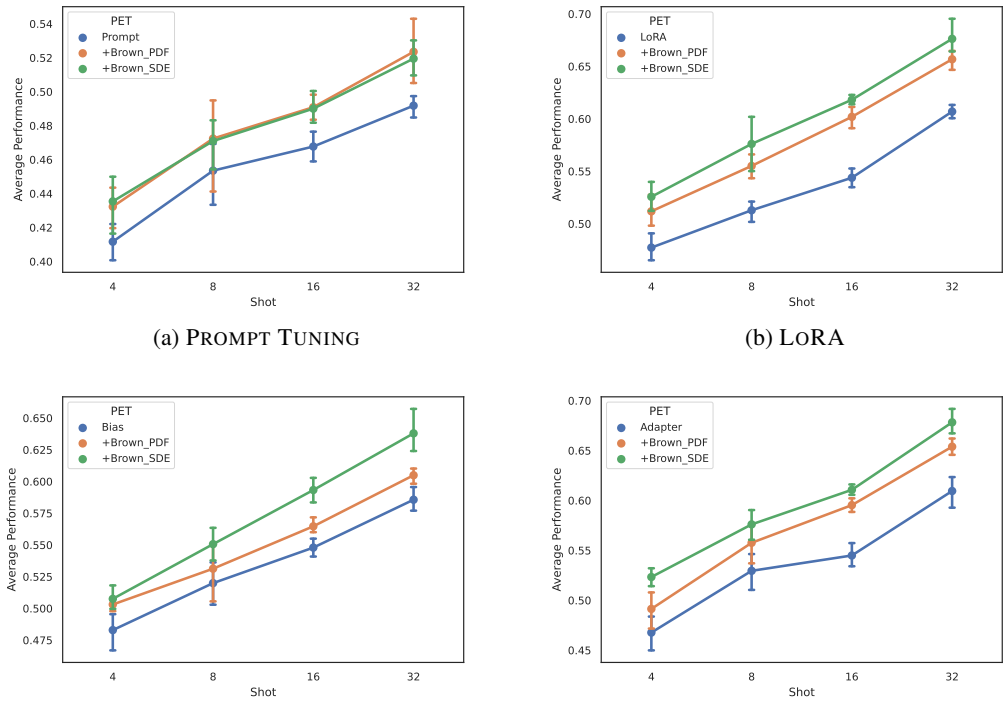
D CALCULATION OF CORRELATION IN SECTION 5

In this section, we elaborate on how we calculate the correlations reported in Table 4 and Table 5.

D.1 CORRELATION BETWEEN NUMBER OF SHOTS AND DISTANCE TO BRIDGE

Definition D.1 (Tie)**.** A pair of observation $\{(x_i, y_i), (x_j, y_j)\}$ is defined as tied if $x_i = x_j$ or $y_i = y_j$.

Since we generate the few-shot datasets using 5 random seeds for each shot, each PET has 5 results for each shot. This results in observations with ties, e.g., the two observations for distances on the first seed and second seed for 8-shot $\{(8, d_1), (8, d_2)\}$ are tied. To calculate the correlation for data with ties, the Tau-b of Kendall's rank correlation is more suitable than Pearson's correlation. We



(a) PROMPT TUNING (b) LORA

(c) BITFIT (d) ADAPTER

Figure 4: The average $\text{Deberta}_{\text{xlarge}}$ few-shot GLUE results trained with different PETs under different shots. The results are averaged across 5 different seeds and the error bars indicate the 95% confidence.

therefore report the Kendall's rank correlation for the correlation between the number of shots and the hidden states distance to the latent bridges.

D.2 CORRELATION BETWEEN PERFORMANCE AND DISTANCE TO BRIDGE

We mix all the few-shot results for different shots and different seeds to form observations of performance and the hidden states distances to bridges, and then calculate the Pearson's correlation.

E HYPER-PARAMETERS

E.1 TRAINING g_γ

we use simple 3-layer MLP for g_γ in all of our experiments. For PDF regularizer, the output dimensions of each layer are 1024, 256, 128, and for SDE regularizer, the output dimensions of each layer are 1024, 256, 32. We observe that the running time increases noticeably with the final output dimension for SDE regularizer, we thus choose a smaller one for SDE regularizer. The hyper-parameters for training g_γ are listed in Table 10.

E.2 TRAINING PETS ON FULL-SET GLUE

We run all the experiments for 50k steps, and evaluate on the development set every 1k steps. For $\text{BERT}_{\text{large}}$, we use 32 as the batch size while for $\text{Deberta}_{\text{xlarge}}$, we use 16 as the batch size. We choose learning rate 1e-3 for Prompt tuning for both PLMs, and 1e-4 for other PETs for both PLMs. We use 0.01 weight decay, 1.0 maximum gradient norm and no learning rate warm-up for all the experiments. We search the best regularization strength α in $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for PDF regularizer, and in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2,$

Table 6: The complete results on GLUE for $\text{BERT}_{\text{large}}$ under 16-shot setting. We exclude CoLA because all PETs fail to give reasonable result in few-shot setting.

PET	MNLI	QQP	QNLI	SST-2	MRPC	RTE	Average	Δ
PROMPT	$38.1_{1.5}$	$53.0_{3.1}$	$51.6_{1.4}$	$70.1_{4.9}$	$50.1_{3.0}$	$48.0_{1.3}$	$51.8_{0.9}$	-
+BROWN_PDF	$38.7_{2.3}$	$54.9_{2.8}$	$52.1_{1.1}$	$75.0_{11.0}$	$52.8_{2.2}$	$50.8_{3.3}$	$54.0_{1.8}$	2.2
+BROWN_SDE	$\mathbf{40.6}_{0.8}$	$55.4_{2.1}$	$52.9_{1.6}$	$80.0_{10.9}$	$51.9_{3.6}$	$51.7_{3.1}$	$55.4_{1.5}$	3.6
+OU_PDF	$38.9_{1.6}$	$54.9_{2.5}$	$51.8_{2.0}$	$71.4_{9.3}$	$\mathbf{54.3}_{2.6}$	$51.5_{3.5}$	$53.8_{1.9}$	2.0
+OU_SDE	$40.0_{1.7}$	$\mathbf{56.8}_{1.5}$	$\mathbf{53.5}_{2.2}$	$\mathbf{82.7}_{3.2}$	$52.7_{2.1}$	$\mathbf{52.0}_{1.9}$	$\mathbf{56.3}_{0.5}$	**4.5**
LORA	$48.7_{4.5}$	$59.9_{5.5}$	$53.2_{1.2}$	$90.2_{1.1}$	$53.6_{3.4}$	$64.2_{0.9}$	$61.6_{0.6}$	-
+BROWN_PDF	$52.0_{1.3}$	$62.7_{2.2}$	$55.1_{3.8}$	$91.3_{0.1}$	$57.4_{5.0}$	$65.3_{1.5}$	$64.0_{0.7}$	2.4
+BROWN_SDE	$\mathbf{54.1}_{0.9}$	$\mathbf{65.5}_{1.4}$	$64.3_{5.1}$	$91.2_{0.3}$	$60.2_{3.1}$	$65.8_{1.4}$	$\mathbf{66.8}_{0.6}$	**5.2**
+OU_PDF	$51.6_{2.4}$	$62.5_{2.3}$	$55.3_{3.4}$	$\mathbf{91.4}_{0.5}$	$56.9_{5.2}$	$65.6_{0.4}$	$63.9_{1.1}$	2.3
+OU_SDE	$52.9_{2.6}$	$63.1_{1.5}$	$\mathbf{64.4}_{5.1}$	$91.3_{0.3}$	$\mathbf{62.2}_{2.7}$	$\mathbf{66.2}_{1.1}$	$66.7_{0.7}$	5.1
BITFIT	$48.4_{1.6}$	$56.0_{6.1}$	$51.7_{2.5}$	$90.8_{0.8}$	$52.0_{2.3}$	$61.7_{1.4}$	$60.1_{1.1}$	-
+BROWN_PDF	$48.5_{1.9}$	$56.0_{6.0}$	$53.5_{2.0}$	$91.0_{0.4}$	$53.9_{2.3}$	$63.2_{1.6}$	$61.0_{0.8}$	0.9
+BROWN_SDE	$\mathbf{52.3}_{0.5}$	$\mathbf{61.2}_{2.9}$	$58.8_{4.7}$	$90.8_{0.4}$	$54.8_{3.0}$	$\mathbf{63.9}_{2.5}$	$\mathbf{63.6}_{0.8}$	**3.5**
+OU_PDF	$49.0_{1.8}$	$56.1_{6.1}$	$52.8_{2.2}$	$90.8_{0.7}$	$53.5_{2.2}$	$62.5_{1.3}$	$60.8_{0.8}$	0.7
+OU_SDE	$51.8_{1.2}$	$58.4_{2.2}$	$\mathbf{58.9}_{4.1}$	$\mathbf{91.1}_{0.4}$	$\mathbf{56.4}_{3.6}$	$63.5_{1.4}$	$63.4_{0.8}$	3.3
ADAPTER	$47.4_{3.7}$	$57.0_{7.2}$	$55.8_{2.9}$	$91.0_{0.4}$	$55.8_{2.5}$	$62.7_{2.0}$	$61.6_{1.2}$	-
+BROWN_PDF	$49.0_{4.8}$	$58.5_{7.4}$	$56.9_{3.1}$	$91.4_{0.2}$	$57.2_{4.9}$	$63.2_{3.0}$	$62.7_{1.5}$	1.1
+BROWN_SDE	$\mathbf{52.3}_{2.2}$	$62.4_{2.9}$	$\mathbf{64.8}_{4.5}$	$\mathbf{91.9}_{0.4}$	$57.3_{4.1}$	$\mathbf{63.8}_{1.8}$	$\mathbf{65.4}_{1.3}$	**3.8**
+OU_PDF	$48.4_{4.3}$	$58.4_{7.4}$	$57.7_{3.7}$	$91.3_{0.5}$	$57.1_{3.3}$	$63.0_{1.8}$	$62.6_{1.3}$	1.0
+OU_SDE	$48.3_{4.5}$	$\mathbf{62.8}_{1.8}$	$63.8_{4.7}$	$91.1_{0.4}$	$\mathbf{59.8}_{2.3}$	$63.6_{1.9}$	$64.9_{0.6}$	3.3

Table 7: The complete results on GLUE for $\text{BERT}_{\text{large}}$. The values are the average value of the best performances over three different runs, and the subscripts are the standard deviations. The Δ column shows the difference of the average performance between the PETs with and without our regularizers.

PET	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE	Average	Δ
PROMPT	$84.4_{0.1}$	$85.3_{0.3}$	$91.5_{0.1}$	$95.5_{0.1}$	$73.9_{2.4}$	$55.5_{3.4}$	$60.8_{1.5}$	$78.1_{0.6}$	-
+BROWN_PDF	$84.7_{0.2}$	$\mathbf{85.5}_{0.0}$	$91.8_{0.6}$	$95.7_{0.1}$	$75.4_{0.5}$	$56.4_{3.3}$	$61.5_{2.2}$	$78.7_{0.4}$	0.6
+OU_PDF	$84.7_{0.1}$	$85.4_{0.1}$	$91.8_{0.3}$	$95.6_{0.2}$	$76.9_{1.0}$	$57.1_{1.2}$	$60.5_{3.0}$	$78.9_{0.2}$	0.8
+BROWN_SDE	$\mathbf{84.9}_{0.2}$	$85.4_{0.1}$	$91.8_{0.4}$	$\mathbf{95.8}_{0.3}$	$78.8_{1.2}$	$61.4_{2.9}$	$64.7_{1.1}$	$80.4_{0.2}$	2.3
+OU_SDE	$84.7_{0.2}$	$85.3_{0.1}$	$\mathbf{92.1}_{0.3}$	$95.5_{0.2}$	$\mathbf{80.2}_{0.5}$	$\mathbf{61.5}_{3.5}$	$\mathbf{65.9}_{2.7}$	$\mathbf{80.7}_{0.8}$	**2.6**
LORA	$88.8_{0.1}$	$89.2_{0.2}$	$93.5_{0.2}$	$95.5_{0.1}$	$84.6_{0.4}$	$62.8_{1.6}$	$78.9_{1.6}$	$84.8_{0.3}$	-
+BROWN_PDF	$\mathbf{88.9}_{0.1}$	$\mathbf{89.6}_{0.1}$	$\mathbf{93.9}_{0.1}$	$95.6_{0.2}$	$85.1_{0.7}$	$63.7_{0.5}$	$80.0_{0.5}$	$85.2_{0.1}$	0.4
+OU_PDF	$\mathbf{88.9}_{0.2}$	$89.4_{0.1}$	$93.7_{0.1}$	$\mathbf{95.7}_{0.3}$	$86.0_{0.5}$	$63.6_{0.6}$	$80.5_{0.8}$	$85.4_{0.1}$	0.6
+BROWN_SDE	$\mathbf{88.9}_{0.1}$	$89.5_{0.1}$	$93.7_{0.1}$	$\mathbf{95.7}_{0.1}$	$\mathbf{86.5}_{1.2}$	$\mathbf{63.9}_{0.4}$	$\mathbf{80.9}_{0.8}$	$\mathbf{85.6}_{0.1}$	**0.8**
+OU_SDE	$88.8_{0.0}$	$89.5_{0.1}$	$93.7_{0.2}$	$\mathbf{95.7}_{0.3}$	$86.3_{1.2}$	$63.7_{0.6}$	$80.1_{0.9}$	$85.4_{0.1}$	0.6
BITFIT	$\mathbf{87.9}_{0.2}$	$87.6_{0.1}$	$92.7_{0.2}$	$95.6_{0.1}$	$79.4_{2.3}$	$60.2_{0.8}$	$77.0_{1.5}$	$82.9_{0.3}$	-
+BROWN_PDF	$\mathbf{87.9}_{0.1}$	$\mathbf{87.8}_{0.0}$	$\mathbf{93.0}_{0.2}$	$95.7_{0.1}$	$83.1_{0.8}$	$60.3_{0.6}$	$78.3_{0.9}$	$83.7_{0.2}$	0.8
+OU_PDF	$\mathbf{87.9}_{0.1}$	$\mathbf{87.8}_{0.1}$	$\mathbf{93.0}_{0.1}$	$95.7_{0.1}$	$82.6_{0.8}$	$59.8_{1.0}$	$\mathbf{78.8}_{1.4}$	$83.6_{0.4}$	0.7
+BROWN_SDE	$\mathbf{87.9}_{0.2}$	$87.7_{0.0}$	$92.8_{0.1}$	$95.7_{0.1}$	$83.3_{0.8}$	$\mathbf{61.1}_{1.2}$	$77.7_{1.5}$	$\mathbf{83.8}_{0.3}$	**0.9**
+OU_SDE	$\mathbf{87.9}_{0.1}$	$87.6_{0.1}$	$92.7_{0.1}$	$\mathbf{95.8}_{0.1}$	$\mathbf{83.7}_{0.8}$	$60.8_{1.6}$	$77.1_{0.9}$	$83.7_{0.4}$	0.8
ADAPTER	$88.8_{0.1}$	$89.6_{0.3}$	$93.7_{0.1}$	$95.6_{0.1}$	$83.6_{0.1}$	$60.4_{1.2}$	$79.5_{1.2}$	$84.5_{0.3}$	-
+BROWN_PDF	$\mathbf{89.0}_{0.1}$	$89.7_{0.2}$	$93.8_{0.2}$	$\mathbf{95.8}_{0.1}$	$86.5_{1.1}$	$\mathbf{62.6}_{0.7}$	$\mathbf{83.2}_{0.2}$	$\mathbf{85.8}_{0.2}$	**1.3**
+OU_PDF	$88.9_{0.1}$	$89.7_{0.0}$	$93.8_{0.1}$	$\mathbf{95.8}_{0.1}$	$\mathbf{86.8}_{0.6}$	$61.9_{0.2}$	$82.0_{0.6}$	$85.6_{0.1}$	1.1
+BROWN_SDE	$88.9_{0.1}$	$\mathbf{89.8}_{0.1}$	$\mathbf{93.9}_{0.2}$	$\mathbf{95.8}_{0.1}$	$85.9_{0.4}$	$62.3_{1.8}$	$82.2_{0.2}$	$85.5_{0.2}$	1.0
+OU_SDE	$88.9_{0.1}$	$\mathbf{89.8}_{0.1}$	$93.7_{0.1}$	$95.7_{0.1}$	$85.9_{0.7}$	$62.5_{1.2}$	$82.7_{0.5}$	$85.6_{0.2}$	1.1

0.5, 1.0} for SDE regularizer. The best α for PDF regularizer are listed in Table 11, and best α for SDE regularizer are listed in Table 12.

Table 8: The results on GLUE for Deberta$_{\text{xlarge}}$.

PET	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE	Average	Δ
PROMPT	$87.2_{0.1}$	$86.5_{0.1}$	$93.8_{0.1}$	$\mathbf{96.8}_{0.1}$	$75.4_{2.5}$	$64.2_{3.8}$	$78.8_{3.5}$	$83.2_{0.5}$	-
+BROWN_PDF	$\mathbf{87.6}_{0.1}$	$\mathbf{86.8}_{0.3}$	$\mathbf{94.2}_{0.1}$	$\mathbf{96.8}_{0.1}$	$80.3_{2.9}$	$\mathbf{65.5}_{1.3}$	$\mathbf{79.8}_{0.5}$	$84.5_{0.6}$	1.3
+BROWN_SDE	$\mathbf{87.6}_{0.1}$	$\mathbf{86.8}_{0.1}$	$94.0_{0.1}$	$\mathbf{96.8}_{0.1}$	$\mathbf{84.4}_{0.6}$	$64.8_{0.5}$	$79.5_{1.3}$	$\mathbf{84.8}_{0.2}$	**1.6**
LORA	$\mathbf{91.1}_{0.1}$	$90.3_{0.1}$	$95.1_{0.1}$	$96.8_{0.1}$	$88.7_{0.7}$	$68.0_{1.3}$	$83.4_{1.1}$	$87.6_{0.3}$	-
+BROWN_PDF	$\mathbf{91.1}_{0.0}$	$\mathbf{90.5}_{0.0}$	$\mathbf{95.2}_{0.0}$	$\mathbf{97.0}_{0.1}$	$90.1_{0.8}$	$68.6_{0.8}$	$\mathbf{85.9}_{1.3}$	$88.3_{0.2}$	0.7
+BROWN_SDE	$\mathbf{91.1}_{0.1}$	$90.4_{0.0}$	$95.1_{0.0}$	$96.9_{0.2}$	$\mathbf{90.5}_{0.6}$	$\mathbf{69.6}_{1.1}$	$85.6_{0.9}$	$\mathbf{88.5}_{0.1}$	**0.9**
BITFIT	$90.0_{0.1}$	$\mathbf{88.4}_{0.0}$	$\mathbf{95.0}_{0.0}$	$\mathbf{96.6}_{0.1}$	$87.3_{0.6}$	$66.9_{0.2}$	$82.4_{0.6}$	$86.7_{0.1}$	-
+BROWN_PDF	$\mathbf{90.2}_{0.0}$	$88.3_{0.1}$	$\mathbf{95.0}_{0.1}$	$\mathbf{96.6}_{0.1}$	$89.8_{0.5}$	$\mathbf{67.9}_{0.8}$	$82.9_{0.6}$	$87.2_{0.1}$	0.5
+BROWN_SDE	$90.1_{0.1}$	$88.3_{0.0}$	$94.8_{0.0}$	$\mathbf{96.6}_{0.1}$	$\mathbf{90.4}_{0.5}$	$\mathbf{67.9}_{0.4}$	$\mathbf{83.8}_{0.5}$	$\mathbf{87.4}_{0.1}$	**0.7**
ADAPTER	$91.1_{0.1}$	$90.0_{0.1}$	$95.2_{0.0}$	$96.8_{0.2}$	$87.9_{0.5}$	$68.8_{1.8}$	$85.0_{0.6}$	$87.8_{0.2}$	-
+BROWN_PDF	$\mathbf{91.2}_{0.1}$	$90.0_{0.0}$	$\mathbf{95.3}_{0.1}$	$\mathbf{96.9}_{0.2}$	$89.2_{0.8}$	$70.1_{1.0}$	$\mathbf{86.9}_{1.5}$	$88.5_{0.5}$	0.7
+BROWN_SDE	$\mathbf{91.2}_{0.2}$	$\mathbf{90.1}_{0.1}$	$95.2_{0.0}$	$\mathbf{96.9}_{0.2}$	$\mathbf{90.3}_{0.9}$	$\mathbf{70.8}_{1.1}$	$86.3_{1.4}$	$\mathbf{88.7}_{0.4}$	**0.9**

Table 9: The results on GLUE for Deberta$_{\text{xlarge}}$ under 16-shot setting.

PET	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE	Average	Δ
PROMPT	$34.4_{1.2}$	$53.2_{5.1}$	$51.7_{1.7}$	$73.3_{8.7}$	$50.2_{3.2}$	$2.5_{2.5}$	$52.2_{4.9}$	$45.4_{2.1}$	-
+BROWN_PDF	$35.8_{0.9}$	$57.7_{2.4}$	$\mathbf{53.5}_{1.5}$	$\mathbf{87.5}_{3.0}$	$52.2_{1.9}$	$2.8_{2.9}$	$\mathbf{54.1}_{1.8}$	$\mathbf{49.1}_{0.8}$	**3.7**
+BROWN_SDE	$\mathbf{35.9}_{1.6}$	$\mathbf{59.6}_{6.4}$	$53.1_{1.9}$	$82.1_{4.2}$	$\mathbf{55.4}_{1.4}$	$\mathbf{2.9}_{3.0}$	$\mathbf{54.1}_{1.3}$	$49.0_{1.1}$	3.6
LORA	$43.1_{3.6}$	$68.4_{2.9}$	$60.1_{5.3}$	$\mathbf{91.8}_{1.1}$	$57.6_{1.5}$	$3.1_{3.8}$	$56.6_{2.3}$	$54.4_{1.0}$	-
+BROWN_PDF	$\mathbf{52.1}_{3.4}$	$70.2_{2.9}$	$\mathbf{73.3}_{6.3}$	$91.7_{1.1}$	$59.5_{4.5}$	$14.2_{5.6}$	$60.4_{4.2}$	$60.2_{1.2}$	5.8
+BROWN_SDE	$49.6_{4.3}$	$\mathbf{70.6}_{1.5}$	$72.4_{6.0}$	$90.7_{1.2}$	$\mathbf{59.8}_{4.5}$	$\mathbf{28.9}_{1.5}$	$\mathbf{60.6}_{3.1}$	$\mathbf{61.8}_{0.5}$	**7.4**
BITFIT	$41.9_{3.8}$	$67.7_{2.6}$	$60.3_{4.2}$	$\mathbf{91.8}_{0.8}$	$54.9_{2.5}$	$9.4_{2.4}$	$57.6_{2.0}$	$54.8_{0.8}$	-
+BROWN_PDF	$45.2_{3.7}$	$\mathbf{70.3}_{1.2}$	$65.4_{6.7}$	$90.9_{0.8}$	$55.6_{2.1}$	$8.2_{2.5}$	$\mathbf{59.6}_{2.3}$	$56.5_{0.7}$	1.7
+BROWN_SDE	$\mathbf{45.7}_{3.8}$	$69.2_{2.3}$	$\mathbf{69.2}_{6.3}$	$89.7_{1.3}$	$\mathbf{57.8}_{4.0}$	$\mathbf{24.2}_{4.6}$	$59.4_{2.8}$	$\mathbf{59.3}_{1.2}$	**4.5**
ADAPTER	$43.1_{2.9}$	$67.7_{2.7}$	$55.9_{5.3}$	$\mathbf{91.1}_{0.9}$	$56.1_{2.1}$	$8.6_{5.6}$	$59.0_{2.3}$	$54.5_{1.4}$	-
+BROWN_PDF	$\mathbf{50.7}_{3.0}$	$70.1_{1.6}$	$70.9_{5.2}$	$90.6_{1.7}$	$57.6_{4.3}$	$16.1_{7.4}$	$\mathbf{60.6}_{3.9}$	$59.5_{0.8}$	5.0
+BROWN_SDE	$47.1_{1.6}$	$\mathbf{72.0}_{1.1}$	$\mathbf{71.3}_{4.3}$	$91.0_{1.0}$	$\mathbf{59.7}_{4.5}$	$\mathbf{26.4}_{5.5}$	$60.0_{5.7}$	$\mathbf{61.1}_{0.6}$	**6.6**

Table 10: Hyper-parameters for training g_γ

	PDF	SDE
Learning rate	1e-3	1e-3
Weight decay	0	0
Batch size	128	128
Grad norm	1.0	1.0
Max steps	100k	500k
Warmup ratio	0.01	0.01

E.3 TRAINING PETS ON FEW-SHOT GLUE

We run all the experiments for 1k steps, and evaluate on the development set every 50 steps. For all the shots for both regularizers and both models, we use a batch size of 2. Other hyper-parameters are kept the same as in the experiments on full-set GLUE.

Table 11: Best α for PDF regularizer on full-set GLUE

	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE
BERT$_{large}$							
PROMPT	0.05	0.3	0.01	0.2	0.8	0.4	0.4
LORA	0.2	0.8	0.5	0.1	0.1	0.5	0.3
BIAS	0.05	0.3	0.6	0.05	0.8	0.05	0.7
ADAPTER	0.8	0.01	0.7	0.1	0.6	0.4	0.3
Deberta$_{xlarge}$							
PROMPT	0.05	0.1	0.2	0.8	0.7	0.3	0.7
LORA	0.1	0.1	0.05	0.01	0.1	0.01	0.2
BIAS	0.05	0.01	0.01	0.01	0.8	0.01	0.8
ADAPTER	0.3	0.01	0.01	0.01	0.01	0.6	0.1

Table 12: Best α for SDE regularizer on full-set GLUE

	MNLI	QQP	QNLI	SST-2	MRPC	CoLA	RTE
BERT$_{large}$							
PROMPT	0.005	0.005	0.0005	0.0001	1.0	0.2	0.05
LORA	0.0005	0.01	0.01	0.01	0.005	0.0005	0.0005
BIAS	0.001	0.001	0.005	0.0001	1.0	1.0	0.005
ADAPTER	0.001	0.001	0.0005	0.001	0.005	0.5	0.005
Deberta$_{xlarge}$							
PROMPT	0.0001	0.001	0.0001	0.0001	0.2	0.0001	0.0001
LORA	0.0005	0.05	0.0001	0.0001	0.1	0.005	0.0001
BIAS	0.0001	0.0005	0.0001	0.0001	0.05	0.2	0.001
ADAPTER	0.001	0.0005	0.0001	0.0005	0.1	0.001	0.0005