DIFFUSION-INSPIRED RECONFIGURATION OF TRANSFORMERS FOR UNCERTAINTY CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Uncertainty calibration in pre-trained transformers is critical for their reliable deployment in risk-sensitive applications. Yet, most existing pre-trained transformers do not have a principled mechanism for uncertainty propagation through their feature transformation stack. In this work, we propose a diffusion-inspired reconfiguration of transformers in which each feature transformation block is modeled as a probabilistic mapping. Composing these probabilistic mappings reveals a probability path that mimics the structure of a diffusion process, transporting data mass from the input distribution to the pre-trained feature distribution. This probability path can then be recompiled on a diffusion process with a unified transition model to enable principled propagation of representation uncertainty throughout the pre-trained model's architecture while maintaining its original predictive performance. Empirical results across a variety of vision and language benchmarks demonstrate that our method achieves superior calibration and predictive accuracy compared to existing uncertainty-aware transformers.

1 INTRODUCTION

The transformer architecture (Vaswani et al., 2017) has become a universal backbone in most large-scale pre-trained or foundation models spanning numerous domains. These include language (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023), vision (Dosovitskiy et al., 2020; Touvron et al., 2021; Liu et al., 2021), speech (Baevski et al., 2020; Hsu et al., 2021; Radford et al., 2023), and even more complex domains with multi-modal data (e.g., text-image) (Radford et al., 2021; Liu et al., 2023; Driess et al., 2023; Team et al., 2023).

Challenge. Despite their prevalence, existing transformer-based models lack a principled mechanism to assess prediction uncertainty. This often leads to incorrect predictions being assigned high confidence (Guo et al., 2017; Mukhoti et al., 2020) which raises safety concerns in high-stake applications (Moon et al., 2020; Zhu et al., 2023). Uncertainty calibration (UC), or confidence calibration, addresses this issue by ensuring that a model's predicted confidence faithfully reflects its empirical accuracy, thereby making confidence scores reliable for downstream decision-making. For example, effective UC techniques can help determine when to defer to human experts in scenarios where the model exhibits high representation and/or prediction uncertainty, particularly in risk-sensitive applications (Tran et al., 2022a; Rudner et al., 2022b; 2023). While UC has been extensively studied in conventional low-complexity deep neural networks, existing techniques mainly focus on imposing probabilistic priors on network weights and approximating their posteriors via either variational inference or posterior sampling. This quickly becomes both inaccurate and prohibitively expensive when the model complexity increases.

Emerging Paradigm. To sidestep the challenge of computing posteriors over models with exceedingly large complexities, there are emerging approaches that aim to reparameterize the attention outputs as (sparse) Gaussian process predictions (Liu et al., 2020; Chen & Li, 2023; Bui et al., 2025; Chen et al., 2024b) and recast the pre-trained transformer as a probabilistic chain mapping from the data distribution to a feature distribution. This enables principled, uncertainty-aware sampling of feature representations by simulating the chain rather than inferring them via computing the prohibitively expensive model posterior, thereby motivating a more scalable paradigm for UC in large models.

Research Gap. Intuitively, separate reparameterization fails to account for the correlations among feature transformations at different attention blocks that were established during pre-training. This

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

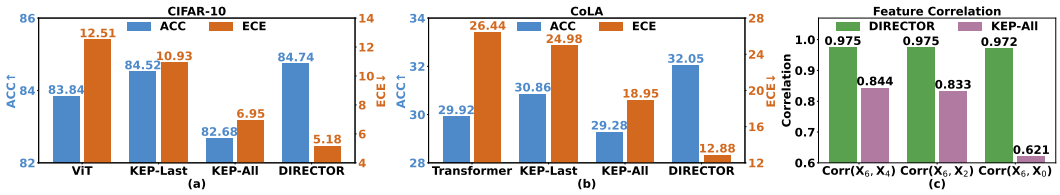


Figure 1: Comparison of accuracy (ACC↑) and uncertainty calibration (ECE↓) across pretrained models (ViT, Transformer), GP-reparameterized method KEP (Chen et al., 2024b) applied to either the last attention block (KEP-last) or all attention blocks (KEP-All), and our method (DIRECTOR) on (a) CIFAR-10 and (b) CoLA dataset. Panel (c) compares the correlation between features at the first layer ($\mathbf{X}_6$) and and those at deeper layers ($\mathbf{X}_4$, $\mathbf{X}_2$, and $\mathbf{X}_0$ at the last layer) for DIRECTOR and KEP-All on CIFAR-10 dataset.

results in an unfavorable exchange between uncertainty calibration and performance. Previous approaches that adopt separate re-parameterization often improve uncertainty calibration at the cost of significantly reduced performance (see Fig. 1a, 1b), which does not align with the expectation that improved uncertainty calibration should, in general, lead to improved performance. From the above intuition, this is not surprising since accurate uncertainty propagation means being able to properly account for the high correlation across feature transformation steps. This is however not upheld in previous approaches according to our investigation. For instance, our results in Fig. 1c show that the correlation between the feature at the first KEP-SVGP's layer and intermediate representations rapidly decreases as it is propagated further towards the solution head. In contrast, our proposed method DIRECTOR manages to correctly preserve this high correlation. As a result, Fig. 1 shows that DIRECTOR indeed improves uncertainty calibration while also improving performance as expected.

Solution Vision. To address this gap, we propose distilling the sequence of reparameterized attention blocks into a unified diffusion model. Rather than treating each block as an independent reparameterization, we model the entire sequence as a continuous stochastic process over the feature embedding space. In this view, the observed transformations of a pre-trained model are interpreted as samples from a diffusion process governed by a single spatiotemporal transition kernel that maps the data distribution to the final representation. This unified view allows us to capture cross-block correlations established during pre-training while providing a principled mechanism for propagating uncertainty.

Technical Contributions. To substantiate this vision, we develop a diffusion-based framework for unified uncertainty propagation across transformer blocks with the following technical contributions:

1. We reinterpret the step-wise feature transformations of a pre-trained transformer as transition samples from a probabilistic path that maps the data distribution to the feature distribution. This perspective generalizes the transformer into a diffusion model parameterized by a unified transition kernel, which can be learned from these observed transitions. Such reconfiguration supports local uncertainty calibration at individual attention blocks while ensuring an accurate flow of uncertainty propagation across the entire network (see Section 2.1).

2. We design a training algorithm that distills the observed sequence of feature transformations into a unified spatiotemporal transition kernel of a diffusion process. The learned kernel captures the inherent correlations among feature transformations across attention blocks established during pre-training, providing a tractable and principled procedure for uncertainty calibration in large pre-trained models. This allows us to establish a generative paradigm for UC, where uncertainty-aware representation samples are drawn directly from the learned diffusion process rather than inferred via intractable model posteriors (see Section 2.2).

3. We conduct extensive experiments on vision and language benchmarks to evaluate calibration quality, robustness, and out-of-distribution (OOD) detection. The results show that our approach consistently improves uncertainty calibration while preserving predictive performance over existing state-of-the-art pre-trained transformer models. Remarkably, it achieves these gains with fewer parameters than the original model, leading to improved memory efficiency. These results demonstrate the feasibility of post-hoc embedding probabilistic reasoning into the internal structure of large pre-trained models for uncertainty calibration without sacrificing performance. This opens a new direction for enhancing their reliability in safety-critical settings (see Section 3).

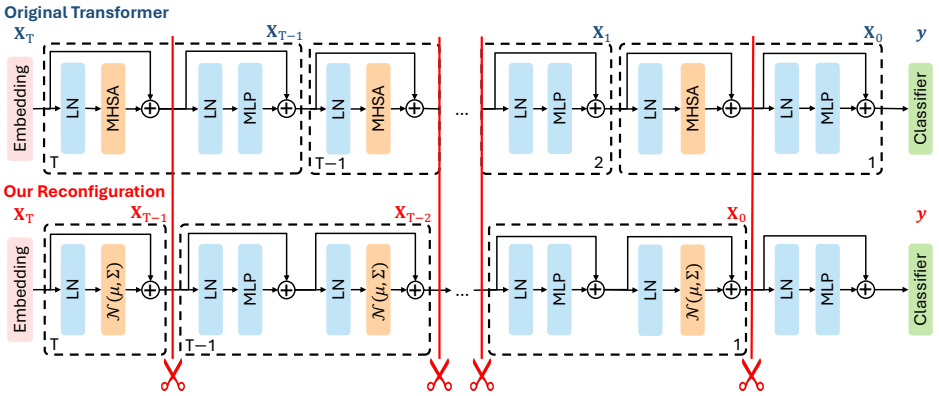


Figure 2: Restructuring a pre-trained transformer such that each block outputs a Gaussian distribution over its intermediate features, effectively aligning its architecture with a probabilistic path.

2 Diffusion-Inspired Reconfiguration of Transformers

Recent efforts to incorporate uncertainty into transformer-based models have uncovered a connection between multi-head self-attention (MHSA) and Gaussian processes (GPs) which shows that the deterministic output of a MHSA block corresponds to the posterior mean of a GP conditioned on its input (Chen & Li, 2023; Bui et al., 2025; Chen et al., 2024b) (see Appendix A.2). Although this insight offers a principled approach for uncertainty calibration at individual attention blocks, propagating it across MHSA blocks remain challenging. The difficulty arises because GP-reparameterized MHSA is interleaved with point-estimated components such as feed-forward networks (MLP) and layer normalization (LN). This disrupts the flow of uncertainty propagation since the interleaved sequence does not align with a well-defined stochastic path with a proper probabilistic transition model, particularly along the point-estimated segments of the pre-trained model.

To enable principled uncertainty propagation in pre-trained transformers, we instead propose a structural reconfiguration that reorganizes the model into a well-defined probabilistic path. In particular, we repartition the architecture such that each transformation block ends with an MHSA block (see Fig. 2), whose output can be interpreted as a Gaussian distribution over intermediate features. This restructuring instead views the aforementioned point-estimated network segments as additional parameterization of a GP-reparameterized MHSA rather than observations of a latent probabilistic transition (see Section 2.1). The resulting sequence of neuralized Gaussian transitions thus becomes well-aligned with the reverse-time stochastic process of a diffusion model.

These neuralized Gaussian transitions can then be viewed as discrete observations at different time steps of the diffusion's reverse-time process. We can thus learn this process via distilling these observed transition across different timesteps into a unified spatiotemporal transition kernel. This can be achieved via adopting variational inference as inspired by score-based diffusion methods (Sohl-Dickstein et al., 2015; Ho et al., 2020). This reveals a novel reconfiguration of pre-trained transformers into uncertainty-aware diffusion processes that interestingly enables UC via learning generative models translating between raw data and predictive features (see Section 2.2).

2.1 Reconfiguring Pre-Trained Transformer as Probability Path

Following prior work on uncertainty-aware transformers (Chen & Li, 2023; Bui et al., 2025; Chen et al., 2024b), the output of a kernelized attention head can be interpreted as the predictive mean at the input queries of a Gaussian process (GP) posterior conditioned on the key-value pairs. This means kernelizing attention transforms the original MHSA mechanism into a GP-based variant that naturally incorporates calibrated uncertainty. Each attention head thus admits a reparameterized Gaussian process (GP) structure which induces a neuralized Gaussian transition,

$$\boldsymbol{F}_t^{(h)} \mid \boldsymbol{U}_t \quad \sim \quad \mathbb{N}\left(\bar{m}_t^{(h)}(\boldsymbol{U}_t),\, \bar{\sigma}_t^{(h)}(\boldsymbol{U}_t)\right) \,, \tag{1}$$

where $\boldsymbol{U}_t$ is the input of the t-th MHSA block whereas $\bar{m}_t^{(h)}(\boldsymbol{U}_t), \bar{\sigma}_t^{(h)}(\boldsymbol{U}_t)$ are neuralized mean and covariance functions for its h-th attention head under the reparameterization design (Appendix A.2).

The individual output representation $\boldsymbol{F}_t^{(h)}$ of each attention head h of the t-th MHSA block can then be aggregated via a linear combination $\boldsymbol{O}_t$ which preserves the (neuralized) Gaussian structure:

$$\boldsymbol{R}_t \;=\; \boldsymbol{O}_t \left[\boldsymbol{F}_t^{(1)}, \boldsymbol{F}_t^{(2)}, \ldots, \boldsymbol{F}_t^{(n)}\right] \quad \Rightarrow \quad \boldsymbol{R}_t \mid \boldsymbol{U}_t \;\sim\; \mathbb{N}\Big(\bar{m}_t(\boldsymbol{U}_t), \bar{\sigma}_t(\boldsymbol{U}_t)\Big)\,, \tag{2}$$

where $\bar{m}_t(\boldsymbol{U}_t) \triangleq [\boldsymbol{O}_t \bar{m}_t^{(1)}(\boldsymbol{U}_t), \ldots, \boldsymbol{O}_t \bar{m}_t^{(n)}(\boldsymbol{U}_t)]$ and $\bar{\sigma}_t(\boldsymbol{U}_t) \triangleq \mathrm{blkdiag}[\boldsymbol{O}_t \bar{\sigma}_t^{(h)}(\boldsymbol{U}_t) \boldsymbol{O}_t^\top]$.

This reparameterization thus reconfigures a pre-trained (point-estimate) MHSA block into probabilistic transition function with a uncertainty structure which can be optimized as in previous methods (Chen & Li, 2023; Bui et al., 2025; Chen et al., 2024b). This provides a principled handle for uncertainty calibration. One can assess the local representation uncertainty via the (learned) predictive variance or generate uncertainty-aware representation samples to propagate downstream. This propagation is however disrupted in existing approaches as mentioned previously due to the interleaving of MHSA with point-estimated components such as MLPs and layer normalization (LN). To elaborate, the transformation from one intermediate representation $\boldsymbol{X}_{t-1}$ to the next $\boldsymbol{X}_t$ interleaves the MHSA mechanism with the MLP and LN mechanisms:

$$\boldsymbol{X}_{t-1} \;=\; \mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t)) + \boldsymbol{Z}_t \quad \text{where} \quad \boldsymbol{Z}_t \;=\; \mathrm{MHSA}\left(\mathrm{LN}(\boldsymbol{X}_t)\right) \;+\; \boldsymbol{X}_t\,, \tag{3}$$

where we number transformer block in reverse such that the first transformer block is indexed with $t = T$ and the last is indexed with $t = 1$, as illustrated in the upper part of Fig. 2

To propagate uncertainty under this partition, the point-estimated network segment $\mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t))$ can be viewed as an observed function sampled from some function prior. However, unlike the MHSA which can be viewed as a sampled function from a learnable Gaussian process (GP) prior as established in prior works, it remains unclear how to parameterize and learn a function prior for $\mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t))$ without the risk of prior misspecification. Otherwise, treating it as a deterministic transition collapses the uncertainty structure and consequently disrupts uncertainty propagation.

To sidestep this technical challenge, we propose to instead view $\mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t))$ as an additional parameterization of the neuralized Gaussian transition induced by the GP-reparameterized MHSA. This can be achieved via a rearrangement of transformer's computation blocks as detailed below:

$$\boldsymbol{X}_{t-1} \;=\; \mathrm{MHSA}(\mathrm{LN}(\boldsymbol{Z}_t)) \;+\; \boldsymbol{Z}_t \;\text{ where }\; \boldsymbol{Z}_t \;=\; \begin{cases} \mathrm{MLP}\left(\mathrm{LN}(\boldsymbol{X}_t)\right) + \boldsymbol{X}_t, & \text{if } t \neq T \\ \boldsymbol{X}_T, & \text{otherwise} \end{cases}. \tag{4}$$

This reconfiguration guarantees that each re-partitioned computation block terminates with an MHSA module (see the lower part of Fig. 2). The deterministic transition $\mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t))$ now become additional parameters of the MHSA which can be reparameterized into a neuralized Gaussian transition. Note that the skip connection does not break Gaussianity but only shifts the mean. Consequently, this construction induces a stochastic process $\{\mathbf{X}_t\}_{t=0}^T$ with Gaussian transitions:

$$p\Big(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t\Big) = \mathbb{N}\Big(\boldsymbol{X}_{t-1} \mid m_t(\boldsymbol{X}_t),\, \sigma_t(\boldsymbol{X}_t)\Big), \tag{5}$$

where $m_t(\boldsymbol{X}_t) = \bar{m}_t(\mathrm{LN}(\boldsymbol{Z}_t)) + \boldsymbol{Z}_t$ and $\sigma_t(\boldsymbol{X}_t) = \bar{\sigma}_t(\mathrm{LN}(\boldsymbol{Z}_t))$ with $\boldsymbol{Z}_t$ is defined in Eq. 4. These separately parameterized Gaussian transitions across timesteps can be distilled into a unified spatiotemporal Gaussian transition that defines the reverse-time process of a diffusion model as discussed in Section 2.2. This unified parameterization enables seamless uncertainty propagation while explicitly encoding transition correlations across steps as desired.

Remark. We note that the above reconfiguration does not alter the pre-trained computation, but changes how the point-estimated segment $\mathrm{MLP}(\mathrm{LN}(\boldsymbol{Z}_t))$ is interpreted. Instead of being treated as an observed function drawn from an unknown prior, it is parameterized as part of a Gaussian transition, revealing a learnable representation that is more amenable to uncertainty propagation. For brevity, we also abuse the notation $\boldsymbol{X}/\boldsymbol{F}$ to denote $\mathrm{vec}(\boldsymbol{X})/\mathrm{vec}(\boldsymbol{F})$.

2.2 Distilling Transformer-based Probability Path on Diffusion Model

Under our proposed reconfiguration in Section 2.1, the transformer induces a stochastic path $\{\mathbf{X}_t\}_{t=0}^T$ with Gaussian transitions (Eq. 5), which closely resembles a reverse diffusion process. This defines a distribution over intermediate features conditioned on the original input embedding $\boldsymbol{X}_T$:

$$p\big(\mathbf{X}_{T-1}, \ldots, \mathbf{X}_0 \mid \boldsymbol{X}_T\big) \;=\; \prod_{t=1}^{T} p\big(\mathbf{X}_{t-1} \mid \mathbf{X}_t\big) \;=\; \prod_{t=1}^{T} \mathbb{N}\Big(\boldsymbol{X}_{t-1} \mid m_t(\boldsymbol{X}_t),\, \sigma_t(\boldsymbol{X}_t)\Big)\,, \tag{6}$$

where the transition probability $p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)$ is modeled independently at each timestep. Calibrating uncertainty in this block-wise, decoupled structure is difficult as it does not capture transition correlation and hence does not generalize across timesteps.

To address this limitation, we require a more parsimonious representation that characterizes the entire sequence of transition models in a unified manner. This can be achieved by re-compiling it into a reverse-time diffusion process with a unified spatiotemporal transition model:

$$q_\theta\big(\mathbf{X}_{T-1}, \ldots, \mathbf{X}_0 \mid \boldsymbol{X}_T\big) \;=\; \prod_{t=1}^{T} q_\theta\big(\mathbf{X}_{t-1} \mid \mathbf{X}_t\big) \;=\; \prod_{t=1}^{T} \mathbb{N}\Big(\boldsymbol{X}_{t-1} \mid m_\theta(\boldsymbol{X}_t),\, \sigma_\theta(\boldsymbol{X}_t)\Big)\,, \tag{7}$$

In particular, the entire sequence of neuralized Gaussian transitions derived from the previously described GP-reparameterized of pre-trained transformer can be absorbed into the reverse-time diffusion with a unified spatiotemporal transition via minimizing the following negative log-likelihood, analogous to score matching in diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020):

$$L(\theta) \;=\; \mathbb{E}_{p(\boldsymbol{X}_0 \mid \boldsymbol{X}_T)}\Big[-\log q_\theta\Big(\boldsymbol{X}_0 \mid \boldsymbol{X}_T\Big)\Big]\,. \tag{8}$$

This negative log-likelihood (NLL) loss admits the following upper-bound via variational inference:

$$L(\theta) \;\leq\; \mathbb{H}\Big(p(\mathbf{X}_0 \mid \mathbf{X}_T)\Big) \;+\; \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{X}_t \mid \mathbf{X}_T)}\Big[D_{\mathrm{KL}}\Big(p(\mathbf{X}_{t-1} \mid \mathbf{X}_t) \parallel q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)\Big)\Big]\,, \tag{9}$$

with proof deferred to Appendix A.4. Since the entropy term $\mathbb{H}\big(p(\mathbf{X}_0 \mid \mathbf{X}_T)\big)$ is independent of θ, optimizing the bound in Eq. 9 reduces to minimizing the Kullback-Leibler (KL) divergence:

$$L_1(\theta) \;=\; \mathop{\mathbb{E}}_{t \,\sim\, \mathbb{U}(1,T)} \; \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p(\mathbf{X}_t \mid \mathbf{X}_T)} \Big[D_{\mathrm{KL}}\Big(p(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t) \parallel q_\theta(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t)\Big)\Big]\,, \tag{10}$$

where $t \sim \mathbb{U}(1, T)$ and $\boldsymbol{X}_t$ is sampled via sampling data $\boldsymbol{X}$ and simulating the corresponding output of the $(T-t)$-th block of the pre-trained transformer. This loss aligns the probability path with a diffusion-style transition kernel while enabling generalization across timesteps. To ensure that the learned uncertainty propagation process maps from data to feature distributions which are informative for downstream prediction, we regularize it with an additional performance loss:

$$L_2(\theta) = \mathop{\mathbb{E}}_{(\boldsymbol{X}, \boldsymbol{y}) \sim \boldsymbol{D}} \; \mathop{\mathbb{E}}_{\boldsymbol{X}_0 \sim q_\theta(\boldsymbol{X}_0 \mid \boldsymbol{X}_T)} \Big[\mathrm{loss}\Big(\boldsymbol{X}_0, \boldsymbol{y}\Big)\Big]\,, \tag{11}$$

where $\boldsymbol{X}$ is sampled from the training dataset $\boldsymbol{D}$ and is embedded with $\boldsymbol{X}_T = \mathrm{embed}(\boldsymbol{X})$. $\boldsymbol{X}_0$ is then sampled via iteratively simulating the current estimate of the probability path $q_\theta(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t)$. The parameterization of the unified spatiotemporal transition model can then be obtained via:

$$\theta \;=\; \arg\min_{\theta} \; \Big\{L_1(\theta) \;+\; L_2(\theta)\Big\}\,, \tag{12}$$

which combines the uncertainty-aware (reconfiguration) loss with the performance-aware loss. For implementation details of the above algorithm, please refer to Appendix A.5.

3 Experiments

This section evaluates the efficacy of our proposed method, DIRECTOR: **D**iffusion-**I**nspired **REC**onfiguration of **T**ransf**OR**mers for Uncertainty Calibration, by reconfiguring existing uncertainty-aware transformers into diffusion-based models and comparing their uncertainty calibration and predictive performance against those of the original versions. We describe our experiment settings in Section 3.1 and report detailed empirical results in Section 3.2.

3.1 Experiment Settings

Datasets. We evaluate DIRECTOR using datasets in computer vision (CV) and natural language processing (NLP). In CV, we use the CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). Each dataset contains 45,000 training, 5000 validation, and 10,000 test images. In NLP, we use the IMDB

Table 1: Performance comparison between pre-trained transformers and their diffusion-inspired reconfiguration using DIRECTOR on in-distribution classification tasks. KEP-k/n denotes a pre-trained transformer using GP-reparameterized architecture (KEP (Chen et al., 2024b)) for the last k attention blocks and standard MHSA for the remaining blocks. Better results are shown in **bold**.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10	ViT	83.84±0.09	4.05±0.11	86.42±0.37	67.13±1.98	12.51±0.20	10.91±0.39	28.03±0.15
	DIRECTOR	**85.67**±0.88	**3.10**±0.45	**87.90**±1.04	**61.92**±1.85	**9.67**±0.62	**6.94**±0.52	**23.54**±1.46
	KEP-1/7	84.52±0.25	3.52±0.11	87.52±0.27	65.14±1.27	10.93±0.26	8.21±0.15	25.80±0.40
	DIRECTOR	**86.51**±0.57	**2.78**±0.16	**88.28**±0.16	**61.85**±1.93	**8.61**±0.49	**6.11**±0.27	**21.90**±0.82
	KEP-7/7	82.68±0.12	4.46±0.05	85.71±0.34	66.90±1.78	6.95±0.36	5.89±0.11	25.90±0.23
	DIRECTOR	**84.74**±0.40	**3.47**±0.18	**87.34**±0.33	**62.88**±1.20	**5.18**±0.33	**4.84**±0.11	**22.48**±0.53
CIFAR-100	ViT	52.94±0.63	22.88±0.64	81.07±0.56	75.96±2.39	30.73±0.61	32.71±0.91	74.72±1.20
	DIRECTOR	**57.79**±0.57	**18.58**±0.37	**82.60**±0.11	**71.50**±1.42	**22.31**±0.49	**22.16**±0.36	**62.57**±0.85
	KEP-1/7	55.74±0.77	20.20±0.64	82.10±0.20	73.82±0.92	27.07±0.71	27.45±0.62	68.54±1.18
	DIRECTOR	**58.78**±1.52	**17.63**±1.08	**82.99**±0.41	**71.40**±1.24	**21.17**±1.03	**21.15**±0.93	**60.96**±1.73
	KEP-7/7	57.06±0.56	19.38±0.60	82.02±0.39	72.78±0.69	**21.31**±3.85	22.41±2.29	63.07±2.11
	DIRECTOR	**60.85**±2.98	**16.35**±2.34	**82.91**±0.67	**71.46**±1.43	21.43±2.09	**20.55**±2.42	**58.91**±4.63
IMDB	Transformer	85.59±0.50	4.73±0.32	80.75±0.55	75.45±1.10	6.96±2.05	3.95±0.44	22.28±1.26
	DIRECTOR	**86.07**±0.61	**4.57**±0.39	**80.84**±0.67	**74.24**±0.87	**5.40**±1.90	**3.60**±0.34	**21.08**±1.32
	KEP-1/5	85.76±0.71	4.54±0.42	81.02±0.70	74.87±0.87	5.51±2.94	3.79±0.51	21.62±1.42
	DIRECTOR	**87.13**±0.19	**4.07**±0.30	**81.55**±0.62	**73.35**±0.14	**3.16**±2.54	**3.24**±0.31	**19.31**±0.97
	KEP-5/5	84.57±0.81	5.48±0.60	79.32±1.25	77.03±1.48	7.83±3.28	5.17±2.13	24.23±2.56
	DIRECTOR	**85.74**±0.34	**4.58**±0.23	**80.95**±0.42	**75.08**±1.15	**2.33**±1.54	**3.36**±0.12	**20.70**±0.65
CoLA	Transformer	29.92±1.17	20.80±1.21	64.22±1.46	90.01±2.84	26.44±1.90	19.66±4.18	55.09±2.68
	DIRECTOR	**31.85**±2.46	**19.74**±1.62	**64.52**±1.71	**89.52**±3.61	**23.94**±0.49	**14.29**±3.01	**50.82**±1.19
	KEP-1/5	30.86±2.03	19.86±2.04	65.18±1.83	**89.10**±2.60	24.98±2.08	16.28±4.35	52.86±2.88
	DIRECTOR	**31.84**±1.88	**19.42**±1.83	**65.54**±1.78	90.37±0.89	**13.77**±6.58	**8.26**±3.75	**43.54**±3.65
	KEP-5/5	29.28±1.21	20.82±1.98	64.47±0.90	89.65±1.04	18.95±5.66	11.83±7.96	48.65±5.84
	DIRECTOR	**32.05**±1.56	**18.69**±1.39	**64.71**±1.37	**89.53**±1.67	**12.88**±5.74	**7.68**±1.70	**42.20**±2.50

dataset (Maas et al., 2011), with 20,000 training, 5,000 validation, and 25,000 test samples; and the CoLA dataset (Warstadt et al., 2019), with 6,355 training, 907 validation, and 1,816 test samples.

Baselines. We compare DIRECTOR with various uncertainty-aware baselines: Temperature Scaling (TS) (Guo et al., 2017), Monte Carlo Dropout (MCD) (Gal & Ghahramani, 2016), Stochastic Variational Deep Kernel Learning (SV-DKL) (Wilson et al., 2016a), Kronecker-Factored Last-Layer Laplace Approximation (KFLLA) (Kristiadi et al., 2020), Sparse Gaussian Process Attention (SGPA) (Chen & Li, 2023), and Kernel-Eigen Pair Sparse Variational Gaussian Processes Attention (KEP-SVGP or KEP for brevity) (Chen et al., 2024b).

Pre-Trained Models. We conduct our uncertainty-aware reconfiguration experiments on two pre-trained architectures which include (i) vanilla transformer-based model and (ii) uncertainty-aware transformer KEP (Chen et al., 2024a). For each architecture, we pre-train a 7-layer vision transformer (ViT (Dosovitskiy et al., 2020)) for experiments on CIFAR-10 and CIFAR-100, and a 5-layer transformer (Vaswani et al., 2017) for CoLA and IMDB. We also evaluate variants of uncertainty-aware transformer KEP (Chen et al., 2024a), where the first $n - k$ attention blocks use standard MHSA and the last k blocks use GP-reparameterized attention; we denote this variant as KEP-k/n.

Unified Transition Model. We implement the unified spatiotemporal transition model in our diffusion-based reparameterization using a single-block DiT (Peebles & Xie, 2022), with embedding dimensions matched to the pre-trained models (see Appendix A.6.13 for full hyperparameters). The model is designed to be smaller than the original backbone for improved memory efficiency, containing only 2.7M parameters versus 6.24M in ViT for vision tasks, and 2.59M versus 3.38M in the text transformer for NLP tasks. Results for alternative transition model architectures are reported in Appendix A.6.7.

Evaluation Metrics. For in-distribution classification, we evaluate predictive performance using accuracy (ACC) for CIFAR-10, CIFAR-100, and IMDB, and Matthew's Correlation Coefficient (MCC) for CoLA. Calibration is assessed with Negative Log-Likelihood (NLL $\times$ 10), Expected Calibration Error (ECE %), and Brier Score (%). Failure prediction is measured using Area Under the Risk-Coverage Curve (AURC %), Area Under the Receiver Operating Characteristic Curve (AUROC %), and False Positive Rate at 95% True Positive Rate (FPR95 %). For out-of-distribution (OOD) robustness, the same metrics are applied to the CIFAR-10-C and CoLA-OOD datasets. OOD

Table 2: Comparison of average performance achieved by the diffusion-based reconfigured KEP (Chen et al., 2024b) produced by DIRECTOR and other uncertainty-aware baselines on in-distribution classification tasks. **Blue** marks the best result across all baselines for a dataset while **brown** denotes the second-best. ↑ indicates that higher values are better, while ↓ indicates that lower values are better.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10	TS	83.84±0.09	3.88±0.10	86.82±0.37	65.99±1.94	9.22±0.36	6.58±0.16	25.50±0.13
	MCD	84.06±0.23	8.65±0.03	86.51±0.32	66.15±0.60	9.47±0.16	8.36±0.32	25.45±0.29
	KFLLA	83.84±0.10	3.91±0.11	86.71±0.45	65.44±1.58	8.18±0.80	**6.09**±0.40	**24.98**±0.59
	SV-DKL	83.23±0.17	4.39±0.18	85.94±0.36	66.96±1.30	11.64±0.81	9.85±1.09	27.97±0.77
	SGPA	75.59±3.63	8.41±2.37	82.65±1.71	71.78±2.73	**1.92**±0.55	7.11±0.95	33.98±4.57
	ViT	83.84±0.09	4.05±0.11	86.42±0.37	67.13±1.98	12.51±0.20	10.91±0.39	28.03±0.15
	KEP	**84.52**±0.25	**3.52**±0.11	**87.52**±0.27	**65.14**±1.27	10.93±0.26	8.21±0.15	25.80±0.40
	DIRECTOR	**84.74**±0.40	**3.47**±0.18	**87.34**±0.33	**62.88**±1.20	**5.18**±0.33	**4.84**±0.11	**22.48**±0.53
CIFAR-100	TS	52.94±0.63	22.34±0.61	**82.29**±0.48	71.65±1.98	**17.06**±0.42	21.57±0.52	64.77±0.95
	MCD	53.49±0.62	22.24±0.56	81.60±0.19	73.02±0.51	25.93±0.37	29.24±0.73	70.02±0.92
	KFLLA	52.27±0.86	23.96±0.78	81.30±0.48	**71.42**±1.92	18.52±5.40	20.89±0.57	66.51±1.66
	SV-DKL	51.03±0.60	24.38±0.43	81.32±0.50	73.99±1.40	25.46±0.72	28.93±0.66	71.90±0.74
	SGPA	52.77±0.52	22.84±0.52	81.65±0.36	72.02±1.74	**10.33**±2.25	**19.10**±0.57	**62.08**±1.04
	ViT	52.94±0.63	22.88±0.64	81.07±0.56	75.96±2.39	30.73±0.61	32.71±0.91	74.72±1.20
	KEP	**57.06**±0.56	**19.38**±0.60	82.02±0.39	72.78±0.69	21.31±3.85	22.41±2.29	63.07±2.11
	DIRECTOR	**60.85**±2.98	**16.35**±2.34	**82.91**±0.67	**71.46**±1.43	21.43±2.09	**20.55**±2.42	**58.91**±4.63
IMDB	TS	85.59±0.50	4.73±0.32	80.75±0.55	75.45±1.10	**2.91**±1.51	**3.41**±0.13	21.04±0.74
	MCD	**85.96**±0.42	**4.40**±0.24	**81.40**±0.55	**74.79**±0.88	4.18±2.03	3.47±0.23	**20.72**±0.82
	KFLLA	85.59±0.50	4.71±0.30	80.82±0.48	75.45±1.11	5.84±2.21	6.93±0.00	21.86±1.19
	SV-DKL	85.69±0.66	5.58±0.79	78.54±2.20	75.32±0.84	8.52±1.57	4.49±0.65	23.10±1.66
	SGPA	85.39±0.36	4.96±0.49	80.04±1.14	76.44±0.96	6.04±1.71	3.96±0.50	22.19±1.06
	Transformer	85.59±0.50	4.73±0.32	80.75±0.55	75.45±1.10	6.96±2.05	3.95±0.44	22.28±1.26
	KEP	85.76±0.71	4.54±0.42	81.02±0.70	74.87±0.87	5.51±2.94	3.79±0.51	21.62±1.42
	DIRECTOR	**87.13**±0.19	**4.07**±0.30	**81.55**±0.62	**73.35**±0.14	**3.16**±2.54	**3.24**±0.31	**19.31**±0.97
CoLA	TS	29.92±1.17	20.84±1.23	64.31±1.44	89.93±2.95	**23.22**±2.99	**11.04**±1.91	**51.70**±3.42
	MCD	30.04±1.02	20.66±1.11	64.53±1.00	89.51±1.35	24.96±1.79	17.83±3.61	53.52±2.59
	KFLLA	29.89±1.14	20.82±1.26	64.22±1.46	89.87±3.24	24.36±2.25	12.16±1.49	52.80±2.85
	SV-DKL	30.07±1.41	22.76±2.28	61.98±3.09	**89.00**±2.55	25.71±1.60	17.96±3.26	54.40±2.13
	SGPA	**31.53**±2.05	20.44±2.60	64.34±1.95	90.79±0.87	26.22±1.51	28.65±7.23	54.08±2.44
	Transformer	29.92±1.17	20.80±1.21	64.22±1.46	90.01±2.84	26.44±1.90	19.66±4.18	55.09±2.68
	KEP	30.86±2.03	**19.86**±2.04	**65.18**±1.83	**89.10**±2.60	24.98±2.08	16.28±4.35	52.86±2.88
	DIRECTOR	**32.05**±1.56	**18.69**±1.39	**64.71**±1.37	89.53±1.67	**12.88**±5.74	**7.68**±1.70	**42.20**±2.50

detection performance is quantified using AUROC % and Area Under the Precision-Recall Curve (AUPR %). All metrics are reported as mean ± standard error over five runs. All experiments are conducted on 8 NVIDIA L40 GPUs.

3.2 Results and Discussion

3.2.1 In-Distribution Classification

Comparison with Pre-Trained Models. DIRECTOR demonstrates superior performance compared to pre-trained models across nearly all tasks and metrics (Table 1). On CIFAR-10, DIRECTOR outperforms KEP-7/7 on every reported metric, with higher accuracy (84.74% versus 82.68%), lower ECE (5.18 versus 6.95), lower NLL (4.84 versus 5.89), and a lower Brier score (22.48 versus 25.90), while also leading in the failure-prediction metrics. On CIFAR-100, although DIRECTOR does not outperform KEP-7/7 in ECE (21.43 versus 21.31), the difference is marginal ($< 0.2\%$), while DIRECTOR achieves significantly higher accuracy (60.85% versus 57.06%) and a lower Brier score. Additionally, significance testing, reported in Appendix A.6.8, indicates that most improvements are statistically significant (p-value < 0.05). These results demonstrate that DIRECTOR not only provides better-calibrated uncertainty estimates but also enhances predictive accuracy, underscoring the effectiveness of our propagation-based model for UC.

Comparison with Existing Uncertainty-Aware Baselines. Beyond vanilla pre-trained models, DIRECTOR also surpasses other uncertainty-aware approaches in both predictive performance and uncertainty calibration across multiple tasks (Table 2). For a fair comparison, DIRECTOR and KEP are configured with optimal settings and evaluated against standard baselines. Overall, DIRECTOR achieves the **highest** performance in 20/28 settings (across 4 datasets and 7 UC/performance metrics) and the **second-highest** in 6/28 settings, establishing a new state-of-the-art in UC.

Table 3: Comparison on CIFAR10-C.

Method	ACC ↑	ECE ↓	NLL ↓	Brier ↓
ViT	**69.67**±0.34	24.30±0.31	23.59±1.00	53.07±0.59
DIRECTOR	68.89±1.44	**22.32**±1.09	**17.71**±1.02	**51.77**±2.39
KEP-1/7	**69.87**±0.45	22.12±0.47	18.54±0.63	50.65±0.90
DIRECTOR	69.29±0.66	**20.98**±0.65	**16.23**±0.60	**50.07**±1.20
KEP-7/7	59.57±0.30	**21.78**±0.59	**17.17**±0.39	60.67±0.72
DIRECTOR	**68.12**±0.26	22.19±0.19	17.70±0.17	**52.27**±0.35

Table 4: Comparison on CoLA OOD.

Method	MCC ↑	ECE ↓	NLL ↓	Brier ↓
Transformer	18.43±3.55	31.99±2.70	23.82±5.16	65.50±4.32
DIRECTOR	**23.06**±4.69	**28.72**±1.87	**17.18**±3.75	**59.91**±2.20
KEP-1/5	19.44±1.94	30.33±1.48	19.67±4.82	61.97±2.25
DIRECTOR	**22.10**±5.49	**17.50**±7.52	**9.34**±4.50	**49.92**±5.80
KEP-5/5	**21.14**±3.48	23.27±6.61	14.25±10.66	55.12±7.78
DIRECTOR	20.20±5.46	**17.49**±5.93	**8.61**±1.99	**48.55**±3.67

Table 5: Comparison of average OOD detection performance in AUROC (%) and AUPR (%) (with reported standard error) achieved by the tested baselines over 5 independent runs.

Method	SVHN		CIFAR-100		LSUN	
	AUROC ↑	AUPR ↑	AUROC ↑	AUPR ↑	AUROC ↑	AUPR ↑
MCD	87.09±8.53	91.46±4.87	76.27±0.35	78.82±0.41	88.41±2.05	91.19±1.51
KFLLA	89.47±9.07	92.92±5.27	77.27±0.42	79.88±0.42	90.77±2.93	92.61±2.24
SVDKL	86.59±6.86	90.78±4.04	75.99±0.74	77.89±1.23	87.81±2.52	90.60±1.91
SGPA	61.57±5.11	74.59±3.50	73.42±1.87	75.93±1.87	67.34±9.77	76.76±5.93
ViT	**87.09**±8.53	**91.46**±4.87	76.27±0.35	78.82±0.41	**88.41**±2.05	**91.19**±1.51
DIRECTOR	83.19±10.94	88.84±6.37	**78.57**±0.94	**81.33**±0.95	83.97±7.19	88.70±4.50
KEP-1/7	75.28±19.12	81.92±13.36	77.93±0.39	80.85±0.46	85.64±4.47	88.98±3.57
DIRECTOR	**90.73**±4.07	**93.21**±2.78	**79.29**±0.50	**82.11**±0.40	**89.38**±3.22	**92.08**±2.37
KEP-2/7	88.25±4.67	91.56±3.14	77.71±0.55	80.58±0.53	88.35±3.62	91.18±2.75
DIRECTOR	**92.14**±5.70	**94.49**±3.59	**79.43**±0.57	**82.15**±0.57	**91.39**±2.74	**93.63**±1.86
KEP-7/7	77.16±1.62	84.09±1.28	76.21±0.41	78.82±0.41	77.01±3.28	82.42±2.73
DIRECTOR	**79.33**±20.83	**85.92**±13.88	**79.11**±0.39	**81.70**±0.38	**86.84**±4.07	**90.05**±3.16

3.2.2 Distribution Shift Robustness

We also assess both the uncertainty calibration and predictive performance of DIRECTOR in scenarios with distribution shifts in both image classification and linguistic acceptability tasks. For vision, we use the CIFAR-10-C dataset, which includes 15 corruption types (e.g., noise, blur) at 5 severity levels (Hendrycks & Dietterich, 2019). For language, we use the CoLA OOD dataset, which assesses novel linguistic structures (Warstadt et al., 2019).On CIFAR-10-C (see Table 3), DIRECTOR substantially improves the robustness of KEP-7/7 in predictive accuracy and Brier score, while remaining comparable in ECE and NLL under distribution shift. It also remains competitive with ViT and KEP-1/7 in predictive performance while improving on calibration metrics. On CoLA OOD (see Table 4), DIRECTOR significantly improves calibration metrics without sacrificing MCC, except when compared to KEP-5/5 which has slightly better MCC but is weaker on UC metrics. These observations consistently demonstrate both the performance robustness and generalization capability of DIRECTOR under scenarios with distribution shifts.

3.2.3 Out-of-Distribution Detection

Uncertainty-aware baselines can also be evaluated in terms of their abilities to distinguish between (i) correctly classified in-distribution samples, (ii) misclassified in-distribution samples, and (iii) out-of-distribution (OOD) samples. To assess this capability, we report the average performance with standard error of DIRECTOR in a number of OOD detection scenarios (see Table 5) using the AUROC/AUPR metrics and two standard methods: (1) Maximum Softmax Probability (Hendrycks & Gimpel, 2017) and Entropy Maximization (Chan et al., 2021). Using CIFAR-10 as the in-distribution dataset, our evaluation on SVHN, CIFAR-100, and LSUN demonstrates that the performance of most pre-trained transformer (using CIFAR-100 data) in most cases can be substantially improved by their corresponding diffusion-based reconfiguration. Notably, the diffusion-inspired reconfiguration of KEP-2/7 produced by DIRECTOR outperforms all baselines as highlighted in blue.

Additional results, including large-scale experiments on ViT-B-16 (86M) and Qwen-2.5 (494M), deep ensembles (Lakshminarayanan et al., 2017), reconfiguration of the SGPA backbone, post-hoc Temperature Scaling applied to every baseline, computational and memory overhead, and loss component ablations are provided in Appendix A.6.

4 Related Work

In safety-critical decision-making applications (e.g., healthcare (Lopez et al., 2023)), models must recognize when their confidence is low to defer decisions to human experts (Pietraszek, 2007; Tran et al., 2022b). However, existing transformers typically ignore uncertainty due to point-estimate designs throughout their stack of neural transformations (Papamarkou et al., 2024). Prior investigation in Bayesian deep learning (BDL) are often restricted to moderate-sized DL architectures (Kendall & Gal, 2017; Gustafsson et al., 2020; Chien & Ku, 2015; Ritter et al., 2021; Tran et al., 2019; Fortuin et al., 2022; Tran et al., 2020; Rudner et al., 2022a; Qiu et al., 2023), limiting scalability to large networks (Papamarkou et al., 2024). To elaborate, we next discuss two main UC paradigms:

UC integration during training. These methods treat model parameters as random variables and learn their posterior distributions conditioned on data. Bayesian neural networks approximate posteriors via MCMC or variational inference (Blundell et al., 2015; Guo et al., 2022), while deep ensembles (Lakshminarayanan et al., 2017) approximate them non-parametrically through diverse initializations. Evidential methods (Wilson et al., 2016b; Sensoy et al., 2018) map features to prior parameters over the likelihood for a closed-form uncertainty estimation via conjugate priors. Sampling-based methods offer higher fidelity by avoiding structural assumptions but incur prohibitive sampling cost for large models (Wenzel et al., 2020). In contrast, variational and evidential approaches are more scalable but less accurate due to biased approximations and restrictive parameterizations (Wilson et al., 2022; Chen et al., 2015). Overall, these approaches approximate or sample from the parameter posterior, which remains highly intractable and often yields unreliable estimates.

UC integration post-training. These methods recalibrate prediction confidence by augmenting a trained model's output without altering most its parameters, including data augmentation (Wang et al., 2019), Monte Carlo dropout (Gal & Ghahramani, 2016), and input-gradient norms (Ash et al., 2019). There are also learning-based approaches that adjust output probabilities to better reflect correctness, such as temperature scaling (Guo et al., 2017), replacing the solution head with probabilistic alternatives (e.g., Gaussian processes (Rasmussen & Williams, 2006), SNGPs (Liu et al., 2020; Bradshaw et al., 2017)), or Laplace approximation that fits a local Gaussian approximation to the weight posterior around the model's learned parameters (He et al., 2026). More recently, conformal prediction (Marx et al., 2022) offers a black-box calibration method that uses a pre-trained model's softmax scores and test-time data to produce prediction sets with marginal coverage guarantees.

Uncertainty calibration versus uncertainty quantification. Beyond the uncertainty calibration methods discussed above, uncertainty quantification also plays an important role in safety-critical decision-making applications. While closely related, uncertainty calibration (UC) and uncertainty quantification (UQ) address different aspects of predictive uncertainty. UC focuses on the reliability of a model's confidence estimates, ensuring that predicted confidence aligns with empirical correctness (e.g., predictions made with confidence p are correct approximately p fraction of the time). In contrast, UQ aims to model and represent uncertainty itself, characterizing predictive variability arising from sources such as data noise (aleatoric uncertainty) and model uncertainty (epistemic uncertainty). As a result, a model may exhibit rich uncertainty estimates but remain poorly calibrated, or be well-calibrated yet provide uninformative uncertainty. This distinction highlights that UC and UQ are complementary but non-interchangeable objectives, and motivates approaches that explicitly target calibration reliability in large-scale transformer models.

5 Conclusion

We introduce a diffusion-inspired reconfiguration of pre-trained transformers that enables principled uncertainty propagation across the entire feature transformation stack. Our approach builds on the established connection between multi-head self-attention (MHSA) and Gaussian process (GP) prediction, reparameterizing the feature transformation stack as a sequence of neuralized Gaussian transitions. This sequence is distilled into a diffusion process with a learnable unified spatiotemporal transition model that maps between data and feature distributions, embedding expressive uncertainty-aware structure within the original transformer while preserving predictive performance. We comprehensively evaluate our method across diverse vision and language tasks, consistently demonstrating its effectiveness. These results point to a new direction for embedding probabilistic reasoning into large pre-trained models, enhancing reliability in risk-sensitive applications.

AI USE STATEMENT

We used generative AI tools solely to support manuscript polishing, including improvements to organization, presentation, and the clarity of experimental reporting. These tools were not used to define the research problem, develop the core technical contributions, or design the experiments. All AI-assisted content was carefully reviewed and verified by the authors for accuracy, consistency, and fidelity to the intended technical meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

Jordan T Ash et al. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Long Minh Bui, Tho Tran Huu, Duy Dinh, Tan Minh Nguyen, and Trong Nghia Hoang. Revisiting kernel attention with correlated gaussian process representation. *arXiv preprint arXiv:2502.20525*, 2025.

Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation, 2021. URL https://arxiv.org/abs/2012.06575.

Tian Chen, Jeffrey Streets, and Babak Shahbaba. A geometric view of posterior approximation. *arXiv preprint arXiv:1510.00861*, 2015. URL https://arxiv.org/abs/1510.00861.

Wenlong Chen and Yingzhen Li. Calibrating transformers via sparse gaussian processes. *arXiv preprint arXiv:2303.02444*, 2023.

Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-attention: Self-attention through asymmetric kernel svd in primal representation. *Advances in Neural Information Processing Systems*, 36, 2023.

Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Self-attention through kernel-eigen pair sparse variational gaussian processes. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=4RqG4K5UwL.

Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan AK Suykens. Self-attention through kernel-eigen pair sparse variational gaussian processes. *arXiv preprint arXiv:2402.01476*, 2024b.

Jen-Tzung Chien and Yuan-Chu Ku. Bayesian recurrent neural network for language modeling. *IEEE transactions on neural networks and learning systems*, 27(2):361–374, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Ratsch, Richard E Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xkjqJYqRJy.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hongji Guo, Hanjing Wang, and Qiang Ji. Uncertainty-guided probabilistic transformer for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1935–1944, 2022.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 318–319, 2020.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *ACM Computing Surveys*, 58(7):1–35, 2026.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *International Conference on Machine Learning*, pp. 5436–5446, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

L. Julian Lechuga Lopez, Tim G. J. Rudner, and Farah E. Shamout. Informative priors improve the reliability of multimodal clinical data classification, 2023. URL https://arxiv.org/abs/2312.00794.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

Charles Marx, Shengjia Zhao, Willie Neiswanger, and Stefano Ermon. Modular conformal calibration. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2206.11468.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *international conference on machine learning*, pp. 7034–7044. PMLR, 2020.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in neural information processing systems*, 33:15288–15299, 2020.

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, Jos'e Miguel Hern'andez-Lobato, Aliaksandr Hubin, Alexander Immer, Theofanis Karaletsos, Mohammad Emtiyaz Khan, Agustinus Kristiadi, Yingzhen Li, Stephan Mandt, Christopher Nemeth, Michael A. Osborne, Tim G. J. Rudner, David Rugamer, Yee Whye Teh, Max Welling, Andrew Gordon Wilson, and Ruqi Zhang. Position: Bayesian deep learning is needed in the age of large-scale AI. In *International Conference on Machine Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:270094572.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Tadeusz Pietraszek. On the use of ROC analysis for the optimization of abstaining classifiers. *Machine Learning*, 68:137–169, 2007. URL https://api.semanticscholar.org/CorpusID:25098146.

Shikai Qiu, Tim G. J. Rudner, Sanyam Kapoor, and Andrew Gordon Wilson. Should we learn most likely functions or parameters? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=9EndFTDiqh.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. *MIT press Cambridge, MA*, 1(3), 2006.

Hippolyt Ritter, Martin Kukla, Cheng Zhang, and Yingzhen Li. Sparse uncertainty representation in deep learning with inducing weights. *Advances in Neural Information Processing Systems*, 34: 6515–6528, 2021.

Tim G. J. Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. Continual learning via sequential function-space variational inference. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18871–18887. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/rudner22a.html.

Tim GJ Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 35: 22686–22698, 2022b.

Tim GJ Rudner, Sanyam Kapoor, Shikai Qiu, and Andrew Gordon Wilson. Function-space regularization in neural networks: A probabilistic perspective. In *International Conference on Machine Learning*, pp. 29275–29290. PMLR, 2023.

Murat Sensoy, Lance Kaplan, and M R Kandemir. Evidential deep learning to quantify classification uncertainty. *arXiv preprint arXiv:1806.01768*, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ba-Hien Tran, Simone Rossi, Dimitrios Milios, and Maurizio Filippone. All you need is a good functional prior for bayesian deep learning. *Journal of Machine Learning Research*, 23:74:1–74:56, 2020. URL https://api.semanticscholar.org/CorpusID:227162611.

Dustin Tran, Mike Dusenberry, Mark Van Der Wilk, and Danijar Hafner. Bayesian layers: A module for neural network uncertainty. *Advances in neural information processing systems*, 32, 2019.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022a.

Dustin Tran, Jeremiah Zhe Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Jessie Ren, Kehang Han, Z. Wang, Zelda E. Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, K. Singhal, Zachary Nado, Joost R. van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, E. Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. Plex: Towards reliability using pretrained large model extensions. *ArXiv*, abs/2207.07411, 2022b. URL https://api.semanticscholar.org/CorpusID:250607941.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guotai Wang et al. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10248–10259. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wenzel20a.html.

Andrew G Wilson, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain, 09–11 May 2016b. PMLR. URL https://proceedings.mlr.press/v51/wilson16.html.

Andrew Gordon Wilson, Pavel Izmailov, Matthew D Hoffman, Yarin Gal, Yingzhen Li, Melanie F Pradier, Sharad Vikram, Andrew Foong, Sanae Lotfi, and Sebastian Farquhar. Evaluating approximate inference in bayesian deep learning. In Douwe Kiela, Marco Ciccone, and Barbara Caputo (eds.), *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, volume 176 of *Proceedings of Machine Learning Research*, pp. 113–124. PMLR, 2022. URL https://proceedings.mlr.press/v176/wilson22a.html.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Openmix: Exploring outlier samples for misclassification detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12074–12083, 2023.

A Appendix / supplemental material

A.1 Multi-Head Self-Attention

Self-Attention. Given an input of the attention layer $\mathbf{U} \in \mathbb{R}^{N\times d}$, where N is the number of data points and d is the embedding dimension, self-attention computes queries, keys, and values via $\mathbf{Q} = \mathbf{U}\mathbf{W}_q$, $\mathbf{K} = \mathbf{U}\mathbf{W}_k$, and $\mathbf{V} = \mathbf{U}\mathbf{W}_v$, with projection matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d\times d_h}$, where d_h is projected dimension. The output of the self-attention is:

$$\mathbf{F} \quad = \quad \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V} = \mathbf{A}_{qk}\mathbf{V}, \tag{13}$$

where attention matrix $\mathbf{A}_{qk} \in \mathbb{R}^{N\times N}$ encodes the pairwise similarity between the queries and keys.

Multi-Head Self-Attention (MHSA). MHSA employs n parallel attention heads, each independently computing a self-attention output $\mathbf{F}^{(h)}$ with corresponding $\mathbf{W}_q^{(h)}, \mathbf{W}_k^{(h)}, \mathbf{W}_v^{(h)} \in \mathbb{R}^{d\times d_h}$ as defined in Eq. 13. To maintain computational efficiency, the dimensionality of each head is typically set to $d_h = d/n$. The outputs from all attention heads are concatenated and subsequently projected back to the input dimension, forming MHSA's output:

$$\mathbf{R} = \mathbf{O}\left[\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \ldots, \mathbf{F}^{(n)}\right] \tag{14}$$

where $\mathbf{O} \in \mathbb{R}^{d\times(n\cdot d_h)}$ is a projection matrix.

A.2 Self-Attention as Gaussian Process Inference

Kernel Attention or K-Attention (Tsai et al., 2019) has extended the attention mechanism by replacing cosine similarity with a general kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to compute pairwise similarities. Specifically, the attention matrix $\mathbf{A}_{qk}$ is replaced by a kernel matrix $\mathcal{K}_{qk}$, where each entry is defined as $[\mathcal{K}_{qk}]_{i,j} = \kappa(\mathbf{U}_{i,:}, \mathbf{U}_{j,:})$. Consequently, the output of the kernel attention mechanism is:

$$\mathbf{F} = \mathcal{K}_{qk}\mathbf{V}, \tag{15}$$

Sparse Gaussian Process Attention (SGPA) (Chen & Li, 2023) leverages the Sparse Variational Gaussian Process (SVGP) framework to approximate posterior variances in the attention mechanism. For each dimension i of attention output, the posterior mean and covariance are given by:

$$\mu_i = \mathcal{K}_{qk}\mathbf{V}_{:,i}\ , \text{and}\ \ \Sigma_i = \mathcal{K}_{qq} + \mathcal{K}_{qk}\left(\mathcal{K}_{kk}^{-1}[\mathcal{S}]_{:,:,i}\mathcal{K}_{kk}^{-1} - \mathcal{K}_{kk}^{-1}\right)\mathcal{K}_{kq} \tag{16}$$

where $\mathcal{S} \in \mathbb{R}^{N\times N\times d_h}$ is a set of variational covariance parameters, optimized via the SVGP evidence lower bound. The output for each dimension i is then sampled using the reparameterization trick:

$$\mathbf{F}_{:,i} = \mu_i + \Sigma_i^{1/2}\cdot\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I) \tag{17}$$

By stacking the results across all d_h output dimensions, the final output of the attention head is:

$$\mathbf{F} = \left[\mu_1 + \Sigma_1^{1/2}\epsilon_1\ ,\ \cdots\ ,\ \mu_{d_h} + \Sigma_{d_h}^{1/2}\epsilon_{d_h}\right] \in \mathbb{R}^{N\times d_h} \tag{18}$$

Kernel-Eigen Pair Sparse Variational Gaussian Processes Attention (KEP-SVGP) (Chen et al., 2024b). The constraint of imposing a symmetric kernel matrix $\mathcal{K}_{qk}$ in **K-Attention** and **SGPA** (requiring $\mathbf{W}_q = \mathbf{W}_k$) can restrict the model's representation capacity by eliminating the inherent asymmetry of the original attention matrix $\mathbf{A}_{qk}$. To overcome this restriction, **KEP-SVGP** employs two Gaussian Processes (GPs), leveraging the symmetry of $\mathcal{K}_{qk}\mathcal{K}_{qk}^\top$ and $\mathcal{K}_{qk}^\top\mathcal{K}_{qk}$. Each attention output dimension is then computed by combining the contributions from the two GPs.

More specifically, building on the KSVD framework and the Primal-Attention formulation (Chen et al., 2023), KEP-SVGP introduces two sets of s-dimensional attention outputs to model the left and right eigenspaces, denoted as $F^e_{[i]} := F^e[:,i]$ and $F^r_{[i]} := F^r[:,i] \in \mathbb{R}^N$ for $i = 1,\ldots,s$, corresponding to the primal features $e(\mathbf{U})$ and $r(\mathbf{U})$ in (Chen et al., 2023), respectively. To model

these outputs, the following SVGP priors are defined based on the induced symmetric kernels $\mathcal{K}_{qk}\mathcal{K}_{qk}^\top$ and $\mathcal{K}_{qk}^\top\mathcal{K}_{qk}$:

$$\begin{pmatrix}\mathbf{F}_i^e\\ \mathbf{u}_i^e\end{pmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{pmatrix}\mathcal{K}_{qk}\mathcal{K}_{qk}^\top & \mathbf{H}_e\Lambda^2\\ \Lambda^2\mathbf{H}_e^\top & \Lambda^2\end{pmatrix}\right), \quad \begin{pmatrix}\mathbf{F}_i^r\\ \mathbf{u}_i^r\end{pmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{pmatrix}\mathcal{K}_{qk}^\top\mathcal{K}_{qk} & \mathbf{H}_r\Lambda^2\\ \Lambda^2\mathbf{H}_r^\top & \Lambda^2\end{pmatrix}\right), \tag{19}$$

where $\Lambda \in \mathbb{R}^{s\times s}$ is a diagonal matrix of the top-s singular values of $\mathcal{K}_{qk}$, and $\mathbf{H}_e, \mathbf{H}_r \in \mathbb{R}^{N\times s}$ contain the corresponding top-s left and right singular vectors, respectively. Using variational distributions $\mathbf{u}_{[i]}^e, \mathbf{u}_{[i]}^r \sim \mathcal{N}(\mathbf{m}_{\mathbf{u},[i]}, S_{\mathbf{uu},[i]})$, closed-form posteriors $q(\mathbf{F}_i^c|\mathbf{U}) = \int q(\mathbf{F}_i^c|\mathbf{u}_i)q(\mathbf{u}_i)\,\mathrm{d}\mathbf{u}_i$ ($c \in \{e, r\}$) are derived as:

$$\begin{aligned} q(\mathbf{F}_i^e|\mathbf{U}) &\sim \mathcal{N}\Big(\underbrace{E_U\Lambda^{-1}\mathbf{m}_{\mathbf{u},[d]}}_{\mu^e:=\mathbf{m}_{[i]}^e}, \underbrace{E_U\Lambda^{-2}S_{\mathbf{uu},[i]}E_U^\top}_{\Sigma^e:=\mathbf{L}_{[i]}^e{\mathbf{L}_{[i]}^e}^\top}\Big),\\ q(\mathbf{F}_i^r|\mathbf{U}) &\sim \mathcal{N}\Big(\underbrace{R_U\Lambda^{-1}\mathbf{m}_{\mathbf{u},[d]}}_{\mu^r:=\mathbf{m}_{[i]}^r}, \underbrace{R_U\Lambda^{-2}S_{\mathbf{uu},[i]}R_U^\top}_{\Sigma^r:=\mathbf{L}_{[i]}^r{\mathbf{L}_{[i]}^r}^\top}\Big), \end{aligned} \tag{20}$$

where $E_U := [e(\mathbf{U}_i), \ldots, e(\mathbf{U}_N)]^\top \in \mathbb{R}^{N\times s}$ and $R_U := [r(\mathbf{U}_i), \ldots, r(\mathbf{U}_N)]^\top \in \mathbb{R}^{N\times s}$ are the projection matrices w.r.t. right and left singular vectors of KSVD in (Chen et al., 2023). The variational parameters are $\mathbf{m}_\mathbf{u} \in \mathbb{R}^{s\times s}$, $S_\mathbf{uu} \in \mathbb{R}^{s\times s\times s}$ with the components i -th defined as $\mathbf{m}_{\mathbf{u},[i]} := \mathbf{m}_\mathbf{u}[:, i] \in \mathbb{R}^s$, and $S_{\mathbf{uu},[i]} := S_\mathbf{uu}[:, :, i] \in \mathbb{R}^{s\times s}$.

The outputs of the two SVGPs are sampled via the reparameterization trick:

$$F_{[i]}^e = \mathbf{m}_{[i]}^e + \mathbf{L}_{[i]}^e\epsilon, \quad F_{[i]}^r = \mathbf{m}_{[i]}^r + \mathbf{L}_{[i]}^r\epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, I_N) \tag{21}$$

To fuse the outputs, two schemes are proposed: Addition ($F_{[i]}^{add} := F_{[i]}^e + F_{[i]}^r \in \mathbb{R}^N$) and Concatenation ($F_{[i]}^{cat} := [F_{[i]}^e; F_{[i]}^r] \in \mathbb{R}^{2N}$). To align with standard Transformer architecture, the s-dimensional attention outputs are linearly projected to the target dimension d_h. The final output $\mathbf{F} \in \mathbb{R}^{N\times d_h}$ is computed as: $\mathbf{F} = \mathbf{F}^{\text{add}}\mathbf{W}^{\text{add}}$ for the addition, $\mathbf{F} = \mathbf{W}_1^{\text{cat}}\mathbf{F}^{\text{cat}}\mathbf{W}_2^{\text{cat}}$ for the concatenation, where the projection matrices are $\mathbf{W}^{\text{add}} \in \mathbb{R}^{s\times d_h}$, $\mathbf{W}_1^{\text{cat}} \in \mathbb{R}^{N\times 2N}$ and $\mathbf{W}_2^{\text{cat}} \in \mathbb{R}^{s\times d_h}$.

A.3 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) transform data into noise through a gradual forward diffusion process and then learn to reverse this transformation. The forward process incrementally adds Gaussian noise to the data $\mathbf{X}_0$ over T steps:

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \sqrt{1-\beta_t}\mathbf{X}_{t-1}, \beta_t\mathbf{I}), \tag{22}$$

where $\{\beta_t\}_{t=1}^T$ controls the noise schedule. This defines a Markov chain that progressively corrupts the data. The true reverse process $p(\mathbf{X}_{t-1}|\mathbf{X}_t)$ is generally intractable but becomes tractable when conditioned on $\mathbf{X}_0$:

$$p(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{X}_0) = \mathcal{N}\big(\mathbf{X}_{t-1}; \tilde{\mu}_t(\mathbf{X}_t, \mathbf{X}_0), \tilde{\beta}_t\mathbf{I}\big), \tag{23}$$

where closed-form expressions for $\tilde{\mu}_t$ and $\tilde{\beta}_t$ follow from Bayes' rule under the forward process. However, $\mathbf{X}_0$ is unknown at test time, the model instead learns a parameterized reverse process that conditions only on $(\mathbf{X}_t, t)$:

$$q_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t) = \mathcal{N}(\mathbf{X}_{t-1}; \mu_\theta(\mathbf{X}_t, t), \Sigma_\theta(\mathbf{X}_t, t)), \tag{24}$$

Learning proceeds by minimizing the variational bound on the negative log-likelihood of the data, which encourages $q_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t)$ to match the true reverse process $p(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{X}_0)$. This is typically implemented via noise prediction (score matching), where the network predicts the injected noise $\boldsymbol{\epsilon}$ instead of μ_θ.

A.4 Derivation of the upper bound of $L(\theta)$

The negative log-likelihood $L(\theta) = \mathbb{E}_{p(\mathbf{X}_0|\mathbf{X}_T)}\left[-\log q_\theta(\mathbf{X}_0 \mid \mathbf{X}_T)\right]$ is upper bounded by

$$L(\theta) \le \mathbb{H}\Big(p(\mathbf{X}_0 \mid \mathbf{X}_T)\Big) + \sum_{t=1}^{T}\mathbb{E}_{p(\mathbf{X}_t|\mathbf{X}_T)}\Big[D_{\mathrm{KL}}\Big(p(\mathbf{X}_{t-1}|\mathbf{X}_t) \parallel q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)\Big)\Big] \tag{25}$$

Proof. From the definition of $L(\theta)$, we have

$$L(\theta) = -\mathbb{E}_{p(\mathbf{X}_0|\mathbf{X}_T)}\left[\log q_\theta(\mathbf{X}_0 \mid \mathbf{X}_T)\right] \tag{26}$$

$$= -\mathbb{E}_{p(\mathbf{X}_0|\mathbf{X}_T)}\left[\log\left(\int q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T) d\mathbf{X}_{1:T-1}\right)\right] \tag{27}$$

$$= -\mathbb{E}_{p(\mathbf{X}_0|\mathbf{X}_T)}\left[\log\left(\int p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T) \frac{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}{p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T)} d\mathbf{X}_{1:T-1}\right)\right] \tag{28}$$

$$= -\mathbb{E}_{p(\mathbf{X}_0|\mathbf{X}_T)}\left[\log\left(\mathbb{E}_{p(\mathbf{X}_{1:T-1}|\mathbf{X}_0,\mathbf{X}_T)}\left[\frac{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}{p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T)}\right]\right)\right] \tag{29}$$

$$\leq -\mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)} \log\left(\frac{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}{p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T)}\right) \quad \text{(Jensen's inequality)} \tag{30}$$

$$= \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)} \log\left(\frac{p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T)}{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}\right) = L_{VLB} \tag{31}$$

Now, we derive the L_{VLB} as follows:

$$L_{VLB} = \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[\log\left(\frac{p(\mathbf{X}_{1:T-1} \mid \mathbf{X}_0, \mathbf{X}_T)}{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}\right)\right] \tag{32}$$

$$= \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[\log\left(\frac{p(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T) p(\mathbf{X}_0 \mid \mathbf{X}_T)}\right)\right] \tag{33}$$

$$= \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[-\log p(\mathbf{X}_0 \mid \mathbf{X}_T) + \log\left(\frac{p(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}{q_\theta(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T)}\right)\right] \tag{34}$$

$$= \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[-\log p(\mathbf{X}_0 \mid \mathbf{X}_T) + \log\left(\prod_{t=1}^{T} \frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right)\right] \tag{35}$$

$$= \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[-\log p(\mathbf{X}_0 \mid \mathbf{X}_T) + \sum_{t=1}^{T} \log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right)\right] \tag{36}$$

$$= \mathbb{H}\Big(p(\mathbf{X}_0 \mid \mathbf{X}_T)\Big) + \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[\log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right)\right] \tag{37}$$

Additionally, we have:

$$\mathbb{E}_{p(\mathbf{X}_{0:T-1}|\mathbf{X}_T)}\left[\log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right)\right] \tag{38}$$

$$= \int p(\mathbf{X}_{0:T-1} \mid \mathbf{X}_T) \log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right) d\mathbf{X}_{0:T-1} \tag{39}$$

$$= \int p(\mathbf{X}_{t-1}, \mathbf{X}_t \mid \mathbf{X}_T) \log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right) d\mathbf{X}_{t-1} d\mathbf{X}_t \tag{40}$$

$$= \int p(\mathbf{X}_t \mid \mathbf{X}_T) p(\mathbf{X}_{t-1} \mid \mathbf{X}_t) \log\left(\frac{p(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}{q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)}\right) d\mathbf{X}_{t-1} d\mathbf{X}_t \tag{41}$$

$$= \mathbb{E}_{p(\mathbf{X}_t|\mathbf{X}_T)}\Big[D_{KL}\Big(p(\mathbf{X}_{t-1} \mid \mathbf{X}_t) \parallel q_\theta(\mathbf{X}_{t-1} \mid \mathbf{X}_t)\Big)\Big] \tag{42}$$

A.5 Practical implementation of the proposed algorithm in Eq.12

The uncertainty-aware transition parameterization can be obtained via the objective in Eq.12:

$$\theta \quad = \quad \arg\min_{\theta} \ \Big\{L_1(\theta) + L_2(\theta)\Big\} \tag{43}$$

which combines the matching loss

$$L_1(\theta) \quad = \quad \mathop{\mathbb{E}}_{(\boldsymbol{X}_t, t)\sim p(\mathbf{X}_t|\mathbf{X}_T)}\Big[\mathrm{D}_{\mathrm{KL}}\Big(p(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t) \parallel q_\theta(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t)\Big)\Big], \tag{44}$$

with the performance-aware loss

$$L_2(\theta) \quad = \quad \mathop{\mathbb{E}}_{(\boldsymbol{X},\boldsymbol{y})\sim\boldsymbol{D}} \; \mathop{\mathbb{E}}_{\boldsymbol{X}_0\sim q_\theta(\boldsymbol{X}_0|\boldsymbol{X}_T)} \Big[\mathrm{loss}\Big(\boldsymbol{X}_0, \boldsymbol{y}\Big)\Big] \,, \tag{45}$$

where $\boldsymbol{X}$ is sampled from the training dataset $\boldsymbol{D}$ and is embedded with $\boldsymbol{X}_T = \mathrm{embed}(\boldsymbol{X})$. $\boldsymbol{X}_0$ is then sampled via iteratively simulating the current estimate of the probability path $q_\theta(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t)$. In addition, the KL divergence in Eq. 44 can be reduced to the following loss,

$$D_{\mathrm{KL}}\Big(p \parallel q_\theta\Big) \quad \propto \quad \frac{1}{2}\left[\mathrm{tr}\left[\sigma_\theta(\boldsymbol{X}_t,t)^{-1}\sigma_t(\boldsymbol{X}_t)\right] + \log\frac{|\sigma_\theta(\boldsymbol{X}_t,t)|}{|\sigma_t(\boldsymbol{X}_t)|}\right] \tag{46}$$

$$+ \quad \frac{1}{2}\Big(m_\theta(\boldsymbol{X}_t,t) - m_t(\boldsymbol{X}_t)\Big)^\top \sigma_\theta(\boldsymbol{X}_t)^{-1}\Big(m_\theta(\boldsymbol{X}_t,t) - m_t(\boldsymbol{X}_t)\Big) \,. \tag{47}$$

However, the KL computation still presents a challenge due to the high dimensionality of the covariance matrix $\sigma_t(\boldsymbol{X}_t)$, which complicates the evaluation of the trace and log-determinant terms. To mitigate this, we follow the approach of KEP by approximating $\sigma_t(\boldsymbol{X}_t)$ using a Cholesky-like factor $\mathbf{L}_t$ such that $\sigma_t(\boldsymbol{X}_t) = \mathbf{L}_t\mathbf{L}_t^\top$. For the learned covariance $\sigma_\theta(\boldsymbol{X}_t,t)$, we employ a diagonal parameterization, whose Cholesky-like factor reduces to the element-wise square root of its diagonal entries. As demonstrated in Appendix A.6.5, this choice is substantially more efficient than a full covariance parameterization without sacrificing performance. Matching these Cholesky-like factors ensures that the corresponding covariance matrices are aligned, effectively nullifying the trace and log-determinant terms in the KL divergence and enabling efficient optimization. Employing Cholesky-like factor and incorporating weighting terms yields the final objective:

$$\theta \quad = \quad \arg\min_\theta \; \Big\{\lambda_{\mathrm{mean}} L_{\mathrm{mean}}(\theta) + \lambda_{\mathrm{Cholesky}} L_{\mathrm{Cholesky}}(\theta) + \lambda_{\mathrm{NLL}} L_2(\theta)\Big\} \,, \tag{48}$$

where λ_{mean}, $\lambda_{\mathrm{Cholesky}}$, and λ_{NLL} are weighting coefficients for the mean matching term, the Cholesky-like factor alignment, and the performance-aware loss, respectively. The individual loss components are defined as follows:

$$L_{\mathrm{mean}}(\theta) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{X}_t|\mathbf{X}_T)}\left[\|m_\theta(\mathbf{X}_t,t) - m_t(\mathbf{X}_t)\|_2^2\right], \tag{49}$$

$$L_{\mathrm{Cholesky}}(\theta) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{X}_t|\mathbf{X}_T)}\left[\left\|\Big(\sigma_\theta(\mathbf{X}_t,t)^{1/2} - \mathrm{Chol}(\sigma_t(\mathbf{X}_t))\Big)\right\|_2^2\right], \tag{50}$$

$$L_2(\theta) = \mathop{\mathbb{E}}_{(\boldsymbol{X},\boldsymbol{y})\sim\boldsymbol{D}} \; \mathop{\mathbb{E}}_{\boldsymbol{X}_0\sim q_\theta(\boldsymbol{X}_0|\boldsymbol{X}_T)} \Big[\mathrm{loss}\Big(\boldsymbol{X}_0, \boldsymbol{y}\Big)\Big], \tag{51}$$

where $\mathrm{Chol}(\cdot)$ denotes the Cholesky-like factorization. We provide the pseudocode for one training iteration of DIRECTOR in Algorithm 1.

A.6 Additional experiments

A.6.1 Large-scale experiment

Table 6: Comparison of average performance achieved a pre-trained ViT-B-16 model (fine-tuned on CIFAR-10) and its reconfigured variant produced by DIRECTOR.

Method	**#params**	**ACC** ↑	**ECE** ↓	**NLL** ↓	**Brier** ↓
ViT-B-16	86M	**97.880**	18.021	2.564	6.843
DIRECTOR	50M	97.170	**1.611**	**1.122**	**4.762**

To stress test DIRECTOR, we further evaluate its scalability on a larger-scale experiment which requires reconfiguring a pre-trained ViT-B-16 model (pre-trained on the large-scale ImageNet data and fine-tuned on CIFAR-10 data). The reported results in Table 6 show that DIRECTOR 's reconfigured ViT-B-16 achieves substantially better uncertainty calibration with an ECE of 1.611 compared to

Algorithm 1 One training iteration of DIRECTOR. The pre-trained backbone f is frozen.

Input: training pair $(\boldsymbol{X}, \boldsymbol{y})$; pre-trained backbone f with T reconfigured blocks; transition model q_θ with parameters θ; loss weights $\lambda_{\text{mean}}, \lambda_{\text{Cholesky}}, \lambda_{\text{NLL}}$; learning rate η.
Output: updated transition-model parameters θ.

Embed input: $\boldsymbol{X}_T \leftarrow \text{embed}(\boldsymbol{X})$. {shared with the backbone}
With f frozen, run a forward pass to obtain the trajectory $\{\boldsymbol{X}_t\}_{t=0}^{T}$ and, at each block, the ground-truth Gaussian transition statistics $\{(m_t(\boldsymbol{X}_t),\, \sigma_t(\boldsymbol{X}_t))\}_{t=1}^{T}$ (Eq. 5).
for $t = 1, \ldots, T$ **do**
    Predict $(m_\theta(\boldsymbol{X}_t, t),\, \sigma_\theta(\boldsymbol{X}_t, t)) \leftarrow q_\theta(\boldsymbol{X}_t, t)$. {single-block DiT}
end for
Roll out $\boldsymbol{X}_0 \sim q_\theta(\boldsymbol{X}_0 \mid \boldsymbol{X}_T)$ and pass through the classification head to obtain logits $\hat{\boldsymbol{y}}$.
$L_{\text{mean}} \leftarrow \frac{1}{T}\sum_{t=1}^{T} \left\| m_\theta(\boldsymbol{X}_t, t) - m_t(\boldsymbol{X}_t) \right\|_2^2$ {Eq. 49}
$L_{\text{Cholesky}} \leftarrow \frac{1}{T}\sum_{t=1}^{T} \left\| \sigma_\theta(\boldsymbol{X}_t, t)^{1/2} - \text{Chol}(\sigma_t(\boldsymbol{X}_t)) \right\|_2^2$ {Eq. 50}
$L_{\text{NLL}} \leftarrow \text{CE}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ {Eq. 51}
$\mathcal{L}(\theta) \leftarrow \lambda_{\text{mean}} L_{\text{mean}} + \lambda_{\text{Cholesky}} L_{\text{Cholesky}} + \lambda_{\text{NLL}} L_{\text{NLL}}$ {Eq. 12}
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$
return θ

Table 7: Comparison of average performance achieved a pre-trained Qwen-2.5 model (fine-tuned on SST-2) and its reconfigured variant produced by DIRECTOR.

Method	#params	ACC ↑	ECE ↓	NLL ↓	Brier ↓
Qwen-2.5	0.494B	93.005	5.864	3.300	12.184
DIRECTOR	0.479B	**93.119**	**5.482**	**2.953**	**12.089**

18.021 of the original ViT-B-16. DIRECTOR also maintains competitive predictive performance (97.17% vs. 97.88%) while requiring significantly fewer parameters (50M vs. 86M). These findings highlight the ability of DIRECTOR to deliver reliable uncertainty calibration even in large and complex transformer architectures, pre-trained on sophisticated and large-scale data.

We further evaluate the scalability of DIRECTOR to a large-scale pre-trained language model by reconfiguring the last three transformer blocks of a Qwen-2.5 model fine-tuned on SST-2; due to the compute cost of a full-stack reconfiguration on a billion-parameter language model, the remaining blocks retain their original deterministic computation. The reported results in Table 7 show that DIRECTOR's reconfigured Qwen-2.5 achieves better uncertainty calibration with an ECE of 5.482 compared to 5.864 of the original Qwen-2.5, together with consistent improvements across the other calibration metrics, reducing NLL from 3.300 to 2.953 and Brier from 12.184 to 12.089. DIRECTOR also slightly improves predictive accuracy (93.119% vs. 93.005%) while requiring fewer parameters (0.479B vs. 0.494B). These findings highlight the ability of DIRECTOR to deliver reliable uncertainty calibration even in large pre-trained language models.

A.6.2 Deep Ensembles Results

We further evaluate the performance of Deep Ensembles (Lakshminarayanan et al., 2017), a simple yet effective method for uncertainty estimation, when combined with pretrained models and DIRECTOR, as reported in Table 8. The results show that DIRECTOR consistently outperforms pretrained models across both CIFAR-10 and IMDB. On CIFAR-10, DIRECTOR achieves notable improvements in predictive accuracy (up to +5.8% over KEP-7/7), while also delivering stronger uncertainty calibration (lower ECE and NLL) and robustness (AURC, AUROC, Brier score). On IMDB, although the baseline Transformer ensemble already achieves strong results, DIRECTOR further improves calibration (ECE, NLL, Brier) and robustness metrics, with the exception of AURC and AUROC where the Transformer baseline performs slightly better. Overall, these findings demonstrate that DIRECTOR works effectively to ensemble settings, providing consistent benefits across architectures, datasets, and evaluation metrics.

Table 8: Performance comparison of Deep Ensembles for in-distribution classification on CIFAR-10 and IMDB. KEP-k/n denotes a pre-trained transformer using GP-reparameterized architecture (KEP (Chen et al., 2024b)) for the last k attention blocks and standard MHSA for the remaining blocks. **Bold** indicates the better performance in each pairwise comparison between a baseline ensemble (ViT, Transformer, KEP) and diffusion-based reconfigured KEP (Chen et al., 2024b) produced by DIRECTOR ensemble.

Dataset	Method	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10	ViT	87.21	2.52	88.84	58.41	2.70	5.14	19.01
	DIRECTOR	**88.82**	**1.91**	**90.09**	**52.77**	**1.72**	**3.69**	**16.38**
	KEP-1/7	87.48	2.33	89.43	56.39	2.30	4.46	18.29
	DIRECTOR	**89.32**	**1.78**	**90.12**	**54.31**	**1.67**	**3.50**	**15.76**
	KEP-2/7	87.09	2.41	89.48	54.07	2.71	4.48	18.69
	DIRECTOR	**89.20**	**1.79**	**90.33**	**51.85**	**1.86**	**3.52**	**15.73**
	KEP-7/7	83.97	3.77	86.90	62.38	3.58	5.04	23.20
	DIRECTOR	**89.73**	**1.70**	**90.17**	**54.72**	**1.62**	**3.49**	**15.27**
IMDB	Transformer	87.31	**3.55**	**82.92**	71.45	2.36	3.04	18.44
	DIRECTOR	**87.45**	3.56	82.83	**70.68**	**1.97**	**3.00**	**18.25**
	KEP-1/5	87.44	3.63	82.34	72.21	1.56	3.01	18.39
	DIRECTOR	**87.85**	**3.44**	**83.20**	**71.10**	**1.25**	**2.89**	**17.62**
	KEP-2/5	87.77	3.51	82.44	73.14	2.85	3.04	18.18
	DIRECTOR	**87.88**	**3.43**	**82.96**	**72.29**	**1.24**	**2.91**	**17.70**
	KEP-5/5	86.49	4.16	81.54	74.07	2.63	3.28	19.72
	DIRECTOR	**87.66**	**3.43**	**82.95**	**72.13**	**2.07**	**2.95**	**18.05**

Table 9: Reconfiguring an SGPA backbone with DIRECTOR across CIFAR-10, IMDB, and CoLA. SGPA rows reproduce the values reported in Table 2. **Bold** indicates the better value within each pairwise comparison.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10	SGPA	75.59±3.63	8.41±2.37	82.65±1.71	71.78±2.73	**1.92**±0.55	7.11±0.95	33.98±4.57
	DIRECTOR	**82.65**±1.16	**4.47**±0.56	**86.00**±0.65	**66.39**±2.34	4.53±0.58	**5.30**±0.29	**25.10**±1.48
IMDB	SGPA	85.39±0.36	4.96±0.49	80.04±1.14	76.44±0.96	6.04±1.71	3.96±0.50	22.19±1.06
	DIRECTOR	**86.16**±0.62	**4.44**±0.44	**81.04**±0.76	**74.67**±1.43	**3.12**±1.43	**3.36**±0.16	**20.30**±0.84
CoLA	SGPA	31.53±2.05	20.44±2.60	64.34±1.95	90.79±0.87	26.22±1.51	28.65±7.23	54.08±2.44
	DIRECTOR	**32.84**±2.66	**19.04**±1.26	**64.67**±1.18	**89.38**±1.46	**17.00**±7.62	**13.43**±6.22	**45.72**±5.09

A.6.3 Reconfiguring the SGPA Backbone

As discussed in Section 3.2, SGPA exhibits a pronounced trade-off between calibration and predictive performance, achieving low ECE on CIFAR-10 (1.92) at the cost of a substantially lower accuracy (75.59%) compared to its ViT and KEP counterparts. A natural question is whether DIRECTOR can mitigate this trade-off by reconfiguring an SGPA backbone in the same way it reconfigures KEP-SVGP and ViT. To this end, we apply our diffusion-based reconfiguration on top of a pre-trained SGPA backbone (Chen & Li, 2023), following the same two-stage protocol used throughout the paper: we first train an SGPA backbone on each dataset and then train a single-block DiT-based transition model on top of the frozen SGPA features, using the same loss-weighting search procedure described in Appendix A.6.12. The resulting performance, alongside the SGPA stage-1 baseline reported in Table 2, is summarized in Table 9.

As shown in Table 9, DIRECTOR achieves higher predictive performance than the SGPA backbone across all three datasets, with a particularly large gain on CIFAR-10 (82.65% versus 75.59% in ACC) and consistent gains on IMDB (86.16% versus 85.39% in ACC) and CoLA (32.84 versus 31.53 in MCC). On CIFAR-10, DIRECTOR improves two of the three calibration metrics (NLL from 7.11 to 5.30, Brier from 33.98 to 25.10) and all three failure-prediction metrics (AURC from 8.41 to 4.47,

Table 10: In-distribution calibration on CIFAR-10 after applying post-hoc Temperature Scaling (TS) to every baseline. **Bold** marks the better value within each pairwise comparison; **blue** marks the best value across the whole table.

Method	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
ViT + TS	83.84±0.09	3.88±0.09	86.82±0.33	65.99±1.73	9.20±0.32	6.57±0.14	25.49±0.12
DIRECTOR + TS	**85.67**±0.88	**3.04**±0.40	**88.27**±0.83	**60.65**±1.23	**5.68**±0.51	**4.76**±0.30	**21.45**±1.22
KEP-1/7 + TS	84.52±0.25	3.43±0.10	87.85±0.23	63.22±0.95	6.97±0.19	5.36±0.08	23.32±0.29
DIRECTOR + TS	**86.51**±0.57	**2.73**±0.15	**88.65**±0.22	**59.89**±0.61	**4.64**±0.26	**4.33**±0.15	**20.01**±0.62

AUROC from 82.65 to 86.00, FPR95 from 71.78 to 66.39), with ECE being the only column where SGPA retains a smaller value (1.92 versus 4.53). Although SGPA achieves the lowest CIFAR-10 ECE, this result reflects an unfavorable accuracy/calibration trade-off rather than genuinely well-calibrated predictions: ECE alone does not measure predictive utility, and on a balanced 10-class dataset such as CIFAR-10 a random guesser that always outputs the uniform distribution is perfectly calibrated in expectation, with confidence 0.1 matching accuracy 0.1. Reading ECE jointly with NLL and Brier, which jointly penalize under-confident predictions and predictive errors, shows that DIRECTOR delivers a substantially better calibration profile overall (NLL 5.30 versus 7.11 and Brier 25.10 versus 33.98) while raising ACC from 75.59% to 82.65%. On IMDB and CoLA, DIRECTOR additionally reduces ECE, NLL, and Brier simultaneously, mirroring the trend observed for the KEP backbone in Table 1. These results demonstrate that the proposed reconfiguration extends naturally to the SGPA backbone, closing the predictive-performance gap that SGPA exhibits in Table 2 while retaining the calibration trade-off pattern observed for ViT and KEP in Section 3.

A.6.4 Effect of Post-hoc Temperature Scaling on Every Baseline

To examine whether the calibration gains reported by DIRECTOR can be explained by a single temperature parameter, we apply post-hoc TS to *every* baseline considered in the in-distribution setting on CIFAR-10, including the GP-aware variants (KEP-k/n and SGPA) that are typically left untouched in the literature. For each baseline, a single temperature scalar is fitted by minimizing the validation NLL via L-BFGS and is then applied to the test-time logits. The resulting in-distribution metrics, averaged over five random seeds and computed on the same checkpoints used to produce Table 1, are summarized in Table 10.

As shown in Table 10, DIRECTOR outperforms each TS-calibrated baseline pairwise on every metric. Within the ViT pair, DIRECTOR + TS improves ACC from 83.84% to 85.67% and ECE from 9.20 to 5.68, while also reducing AURC, NLL, and Brier scores. The same pattern holds within the KEP-1/7 pair, where every metric improves under DIRECTOR, and DIRECTOR (from KEP-1/7) + TS achieves the best value across the whole table on ACC, AURC, AUROC, FPR95, ECE, NLL, and Brier. These results confirm that the improvements reported in Section 3 originate from the diffusion-based reconfiguration itself, and not from the absence of post-hoc calibration on the baselines, and that the calibration improvements offered by DIRECTOR cannot be replicated by simply applying post-hoc TS to existing uncertainty-aware transformers.

A.6.5 Varying covariance structures

The covariance of the pretrained feature distribution, denoted by $\sigma_t(X_t)$, is approximated using a Cholesky-like factorization as employed in KEP-SVGP (Chen et al., 2024b), which we select as the pretrained model for consistency. In addition, we adopt a diagonal parameterization for the learned covariance $\sigma_\theta(X_t, t)$, a design choice that is standard in diffusion models (Ho et al., 2020).

To examine whether this simplification compromises calibration quality, we conduct a small-scale empirical study comparing the diagonal parameterization against another alternative covariance structure, full parameterization. The results, summarized in Table 11, demonstrate that the diagonal approximation not only yields significantly faster training time but also achieves superior calibration performance relative to the more expressive covariance structure. These observations support the efficiency and effectiveness of the diagonal covariance design in our probabilistic reconfiguration framework.

Table 11: Comparison of different covariance structures under two pretrained baselines ViT and KEP-1/7.

Baseline	Covariance Structure	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓	Training time (hours)
ViT	Diagonal	**86.38**	**2.70**	**88.80**	**61.16**	**9.00**	**6.34**	**22.21**	**1.75**
	Full (dimension-wise)	81.55	4.95	85.48	68.08	10.70	7.28	28.88	2.50
	Baseline	83.90	4.02	86.71	65.16	12.57	10.94	27.80	3.85
KEP-1/7	Diagonal	**86.22**	**2.83**	**88.35**	**61.90**	**8.85**	**6.21**	**22.23**	**1.75**
	Full (dimension-wise)	83.80	3.81	87.26	62.10	11.16	8.12	26.34	2.50
	Baseline	84.61	3.52	87.44	65.04	10.86	8.17	25.59	6.51

Table 12: Inference time and memory consumption on CIFAR-10.

Method	Inference Time (s)	Inference Memory (GB)
KEP-7/7	1.56	0.3348
KEP-1/7	1.07	0.2916
SGPA	2.62	0.5768
ViT	0.93	0.1509
DIRECTOR	1.21	0.1040

We additionally report the wall-clock training time of the pretrained baselines, which corresponds to the cost of pre-training the underlying ViT or KEP-1/7 backbone for 300 epochs on CIFAR-10. As shown in the last column of Table 11, the ViT and KEP-1/7 backbones require roughly 3.85 and 6.51 hours respectively, while the unified transition model with the diagonal covariance parameterization is trained on top of either backbone in only 1.75 hours. The diffusion-based reconfiguration step in DIRECTOR is therefore substantially less expensive than the original pretraining stage, supporting its use as a lightweight post-hoc add-on to existing pretrained transformers.

A.6.6 Computational and memory overhead

We evaluate the computational and memory overhead of our method against several baselines on the CIFAR-10 dataset. As reported in Table 12, our approach achieves substantially lower inference memory consumption than all competing methods, while maintaining an inference time comparable to other baselines. Table 14 reports training time and GPU memory consumption on CIFAR-10 using one NVIDIA L40 GPU. The reconfiguration stage requires a modest increase in training time and memory consumption. The reported time for DIRECTOR does not include training its pre-trained backbone.

Inference time with increasing depth. We further evaluate inference time for models with 7, 14, and 21 layers on one L40S GPU. As reported in Table 13, the inference time of DIRECTOR increases approximately linearly with depth, from 1.21 seconds at 7 layers to 2.48 seconds at 21 layers.

Table 13: Inference time in seconds for increasing model depth, measured on one L40S GPU.

Method	7 layers	14 layers	21 layers
KEP-7/7	1.56	2.57	3.61
KEP-1/7	1.07	1.58	2.15
SGPA	2.62	4.82	7.09
ViT	0.93	1.48	2.01
DIRECTOR	1.21	1.84	2.48

A.6.7 Varying unified transition models

Regarding kernel design, although score networks in diffusion models are commonly implemented using multilayer perceptrons (MLPs), our empirical results indicate that a DiT-based architecture (Peebles & Xie, 2022) consistently achieves superior performance. Table 15 reports a comparison between

Table 14: Training time and GPU memory consumption on CIFAR-10 using one NVIDIA L40 GPU. Backbone training and DIRECTOR reconfiguration are reported separately.

Method	Stage	Time (h)	Memory (GB)
ViT	Backbone training	3.85	1.9
KEP-1/7	Backbone training	6.51	2.2
DIRECTOR	Reconfiguration	1.75	3

Table 15: Comparison of unified transition model (UTM) architectures on CIFAR-10 under different pretrained baselines.

Baseline	UTM	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
ViT	MLP	81.17	5.11	85.47	67.39	**1.36**	**5.52**	26.70
	DiT	**86.18**	**2.95**	**87.74**	**63.82**	8.90	6.27	**22.62**
	Baseline	83.90	4.02	86.71	65.16	12.57	10.94	27.80
KEP-7/7	MLP	83.46	4.16	85.95	65.18	**4.99**	**5.34**	24.37
	DiT	**87.37**	**2.43**	**88.99**	**60.97**	8.53	6.07	**20.84**
	Baseline	82.34	4.42	86.45	65.80	5.81	5.59	25.68

MLP and DiT-based score networks on the CIFAR-10 dataset under two different pretrained baselines. While MLP-based score networks exhibit slightly better calibration on certain metrics, they underperform the corresponding baselines on most other evaluation criteria, particularly predictive accuracy. In contrast, the DiT-based architecture improves performance across the majority of metrics without sacrificing stability. Consequently, we adopt DiT as the score network in our framework.

A.6.8 Statistical significance testing

Regarding statistical significance, we report hypothesis testing results for both the CV task (CIFAR-10) and the NLP task (CoLA), corresponding to the results in Table 1. For CoLA, we additionally perform an extensive evaluation with 40 runs and compute paired t-tests across metrics.

As summarized in Table 16, the majority of comparisons yield p-values below 0.05, indicating that the observed improvements of our method over the corresponding baselines are statistically significant. This holds consistently across accuracy, uncertainty-aware metrics, and calibration measures, demonstrating the robustness of our performance gains.

A.6.9 Comparison with Knowledge Distillation

We compare DIRECTOR with knowledge distillation (KD) Hinton et al. (2015). Results are presented in Table 17. We evaluate KD using a pretrained 7-layer ViT as the teacher and a 3-layer ViT or our DiT as the student, trained with KD (temperature scaling = 3), while DIRECTOR use DiT and trained with our loss function. Although KD improves accuracy, its calibration metrics remain higher compared to DIRECTOR, which achieves superior accuracy and calibration. Additionally, our propagation model is highly parameter-efficient, using only 2.7M parameters, less than half of the original model's 6.27M parameters, resulting in improved memory efficiency.

A.6.10 In-Distribution Classification

We consolidate the experimental results from Table 1 and Table 2, and further include additional findings based on our alignment with KEP-2/7 (vision) and KEP-2/5 (language) configurations, presented in Table 18. DIRECTOR demonstrates state-of-the-art performance across nearly all scenarios, achieving the **highest** score in 23 out of 28 settings and ranking **second** in another 23 out of 28. These results highlight the substantial advantage of DIRECTOR over existing baselines, in terms of both predictive accuracy and uncertainty calibration.

Table 16: Significance testing results on CIFAR-10 (DF=4) and CoLA (DF=39). We report mean ± standard error of performance (Perf.) and paired p-values.

Dataset	Method	ACC / MCC ↑		AURC ↓		AUROC ↑		FPR95 ↓		ECE ↓		NLL ↓		Brier ↓	
		Perf.	p	Perf.	p	Perf.	p	Perf.	p	Perf.	p	Perf.	p	Perf.	p
CIFAR-10	ViT	83.84±0.09	–	4.05±0.11	–	86.42±0.37	–	67.13±1.98	–	12.51±0.20	–	10.91±0.39	–	28.03±0.15	–
	DIRECTOR	**85.67±0.88**	0.0094	**3.10±0.45**	0.0054	**87.90±1.04**	0.0255	**61.92±1.85**	0.0104	**9.67±0.62**	0.0008	**6.94±0.52**	0.0001	**23.54±1.46**	0.0020
	KEP-1/7	84.52±0.25	–	3.52±0.11	–	87.52±0.27	–	65.14±1.27	–	10.93±0.26	–	8.21±0.15	–	25.80±0.40	–
	DIRECTOR	**86.51±0.57**	0.0051	**2.78±0.16**	0.0034	**88.28±0.16**	0.0065	**61.85±1.93**	0.0149	**8.61±0.49**	0.0020	**6.11±0.27**	0.0003	**21.90±0.82**	0.0020
	KEP-7/7	82.68±0.12	–	4.46±0.05	–	85.71±0.34	–	66.90±1.78	–	6.95±0.36	–	5.89±0.11	–	25.90±0.23	–
	DIRECTOR	**84.74±0.40**	0.0000	**3.47±0.18**	0.0000	**87.34±0.33**	0.0001	**62.88±1.20**	0.0023	**5.18±0.33**	0.0000	**4.84±0.11**	0.0000	**22.48±0.53**	0.0000
CoLA	Transformer	29.92±1.06	–	20.80±1.09	–	64.22±1.33	–	90.01±2.58	–	26.44±1.72	–	19.66±3.79	–	55.09±2.43	–
	DIRECTOR	**31.97±2.26**	0.0000	**19.87±1.46**	0.0004	64.52±1.44	–	**89.94±2.02**	0.8877	**23.33±4.51**	0.0001	**15.30±5.62**	0.0001	51.04±4.80	0.0000
	KEP-1/5	32.40±2.32	–	19.94±1.66	–	64.42±2.36	–	90.21±2.12	–	22.45±4.94	–	15.24±5.89	–	50.27±4.65	–
	DIRECTOR	**31.41±2.05**	0.0429	**18.91±1.90**	0.0098	**65.45±1.72**	0.0269	**89.58±1.61**	0.1484	**17.47±8.87**	0.0027	**11.87±5.96**	0.0128	**46.62±6.91**	0.0069
	KEP-5/5	31.36±2.73	–	19.80±1.92	–	64.66±0.98	–	90.26±1.15	–	19.87±5.69	–	14.74±8.28	–	48.69±5.57	–
	DIRECTOR	**31.53±1.95**	0.7545	**18.58±2.49**	0.0121	**65.33±1.93**	0.0361	**88.70±1.63**	0.0000	**15.74±8.57**	0.0139	**10.80±5.91**	0.0174	**44.76±6.98**	0.0076

Table 17: Performance comparison between knowledge distillation methods and our method on CIFAR-10 and CIFAR-100. Mean ± std are reported over five trials.

Method		#params	CIFAR-10				CIFAR-100			
			ACC ↑	ECE ↓	NLL ↓	Brier ↓	ACC ↑	ECE ↓	NLL ↓	Brier ↓
Teacher	ViT (7 layers)	6.27M	83.84±0.09	12.51±0.20	10.91±0.39	28.03±0.15	52.94±0.63	30.73±0.61	32.71±0.91	74.72±1.20
Student	ViT (3 layers)	2.75M	82.20±0.80	10.92±1.53	7.90±0.80	28.58±1.20	54.86±0.65	30.12±0.39	31.25±0.53	72.17±0.78
	DiT	2.70M	84.77±0.63	11.93±0.51	10.26±0.52	26.55±1.17	57.56±0.37	27.37±0.62	27.10±0.77	67.06±0.83
DIRECTOR		2.70M	**85.67**±0.88	**9.67**±0.62	**6.94**±0.52	**23.54**±1.46	**57.79**±0.57	**22.31**±0.49	**22.16**±0.36	**62.57**±0.85

A.6.11 Distribution shift robustness

Additional experiments evaluating the distributional robustness of DIRECTOR are presented in Tables 19,20, and Figure3. These results include evaluations of DIRECTOR aligned with KEP-2/5 and KEP-2/7 configurations. Notably, the TS baseline is excluded from these comparisons, as it is specifically tailored for in-distribution tasks.

Figure 3 illustrates a calibration comparison between pre-trained models (ViT, KEP) and DIRECTOR on the CIFAR-10-C dataset, across five corruption severity levels averaged over 15 corruption types. DIRECTOR demonstrates competitive, and in many cases superior, predictive accuracy, particularly when aligned with KEP-7/7. In terms of uncertainty calibration, DIRECTOR exhibits significantly improved calibration, achieving lower values in ECE, NLL, and Brier score compared to pre-trained baselines.

Table 19 summarizes performance averaged over the 15 corruptions and five severity levels on CIFAR-10-C. While maintaining competitive predictive accuracy with ViT and KEP, DIRECTOR substantially outperforms them in calibration metrics. When aligned with KEP-2/7, our best-performing configuration of DIRECTOR achieves both competitive accuracy and slightly improved calibration compared to other baselines. Additionally, DIRECTOR can be enhanced by integrating post-training baselines such as MCD or aligning with attention-modified methods like SGPA to further improve calibration. However, due to computational constraints, we leave these extensions for future investigation.

Finally, Table 20 compares DIRECTOR against pre-trained models (Transformer and KEP) and other baselines on the CoLA OOD dataset. DIRECTOR achieves the highest MCC score, outperforming all baselines, and shows significant gains in both calibration and failure prediction metrics.

A.6.12 Effect of Varying Loss Weights

We perform extensive ablation studies on the loss weight configurations to investigate their impact on DIRECTOR's performance on both the CoLA and CIFAR-10 datasets. Specifically, we examine how varying the weights assigned to the mean matching term (λ_{mean}), the Cholesky-like factor alignment (λ_{Chol}), and the performance-aware loss (λ_{NLL}) affects both predictive accuracy and uncertainty calibration. When applying diffusion-based reconfiguration with DIRECTOR to KEP-5/5 on CoLA (see Table 21), we observe distinct trends: excessively high λ_{mean}, as in the configuration $(0.9, 0.05, 0.05)$, produces very strong calibration while maintaining decent predictive

Table 18: Performance comparison for in-distribution classification across four tasks. KEP-k/n denotes a pre-trained transformer using GP-reparameterized architecture (KEP (Chen et al., 2024b)) for the last k attention blocks and standard MHSA for the remaining blocks. **Bold** indicates the better performance in each pairwise comparison between a pre-trained model (ViT, Transformer, KEP) and diffusion-based reconfigured KEP (Chen et al., 2024b) produced by DIRECTOR . **Blue** marks the best result across all baselines for a dataset, and **brown** denotes the second-best.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10	TS	83.84±0.09	3.88±0.10	86.82±0.37	65.99±1.94	9.22±0.36	6.58±0.16	25.50±0.13
	MCD	84.06±0.23	8.65±0.03	86.51±0.32	66.15±0.60	9.47±0.16	8.36±0.32	25.45±0.29
	KFLLA	83.84±0.10	3.91±0.11	86.71±0.45	65.44±1.58	8.18±0.80	6.09±0.40	24.98±0.59
	SV-DKL	83.23±0.17	4.39±0.18	85.94±0.36	66.96±1.30	11.64±0.81	9.85±1.09	27.97±0.77
	SGPA	75.59±3.63	8.41±2.37	82.65±1.71	71.78±2.73	**1.92**±0.55	7.11±0.95	33.98±4.57
	ViT	83.84±0.09	4.05±0.11	86.42±0.37	67.13±1.98	12.51±0.20	10.91±0.39	28.03±0.15
	DIRECTOR	**85.67**±0.88	**3.10**±0.45	**87.90**±1.04	**61.92**±1.85	**9.67**±0.62	**6.94**±0.52	**23.54**±1.46
	KEP-1/7	84.52±0.25	3.52±0.11	87.52±0.27	65.14±1.27	10.93±0.26	8.21±0.15	25.80±0.40
	DIRECTOR	**86.51**±0.57	**2.78**±0.16	**88.28**±0.16	**61.85**±1.93	**8.61**±0.49	**6.11**±0.27	**21.90**±0.82
	KEP-2/7	84.32±0.75	3.65±0.22	87.17±0.35	65.30±1.35	11.11±0.46	8.43±0.26	26.20±1.08
	DIRECTOR	**86.62**±0.18	**2.70**±0.07	**88.52**±0.36	**60.70**±2.31	**8.54**±0.14	**5.96**±0.14	**21.59**±0.25
	KEP-7/7	82.68±0.12	4.46±0.05	85.71±0.34	66.90±1.78	6.95±0.36	**5.89**±0.11	25.90±0.23
	DIRECTOR	**84.74**±0.40	**3.47**±0.18	**87.34**±0.33	**62.88**±1.20	**5.18**±0.33	**4.84**±0.11	**22.48**±0.53
CIFAR-100	TS	52.94±0.63	22.34±0.61	82.29±0.48	71.65±1.98	**17.06**±0.42	21.57±0.52	64.77±0.95
	MCD	53.49±0.62	22.24±0.56	81.60±0.19	73.02±0.51	25.93±0.37	29.24±0.73	70.02±0.92
	KFLLA	52.27±0.86	23.96±0.78	81.30±0.48	71.42±1.92	18.52±5.40	20.89±0.57	66.51±1.66
	SV-DKL	51.03±0.60	24.38±0.43	81.32±0.50	73.99±1.40	25.46±0.72	28.93±0.66	71.90±0.74
	SGPA	52.77±0.52	22.84±0.52	81.65±0.36	72.02±1.74	**10.33**±2.25	**19.10**±0.57	62.08±1.04
	ViT	52.94±0.63	22.88±0.64	81.07±0.56	75.96±2.39	30.73±0.61	32.71±0.91	74.72±1.20
	DIRECTOR	**57.79**±0.57	**18.58**±0.37	**82.60**±0.11	**71.50**±1.42	**22.31**±0.49	**22.16**±0.36	**62.57**±0.85
	KEP-1/7	55.74±0.77	20.20±0.64	82.10±0.20	73.82±0.92	27.07±0.71	27.45±0.62	68.54±1.18
	DIRECTOR	**58.78**±1.52	**17.63**±1.08	**82.99**±0.41	**71.40**±1.24	**21.17**±1.03	**21.15**±0.93	**60.96**±1.73
	KEP-2/7	55.82±1.12	20.14±0.94	82.16±0.32	73.34±1.05	27.37±0.55	27.74±0.51	68.65±1.42
	DIRECTOR	**59.45**±1.06	**17.10**±0.80	**83.27**±0.33	**69.94**±1.33	**21.44**±0.86	**20.85**±0.49	**60.15**±1.16
	KEP-7/7	57.06±0.56	19.38±0.60	82.02±0.39	72.78±0.69	**21.31**±3.85	22.41±2.29	63.07±2.11
	DIRECTOR	**60.85**±2.98	**16.35**±2.34	**82.91**±0.67	**71.46**±1.43	21.43±2.09	**20.55**±2.42	**58.91**±4.63
IMDB	TS	85.59±0.50	4.73±0.32	80.75±0.55	75.45±1.10	**2.91**±1.51	3.41±0.13	21.04±0.74
	MCD	85.96±0.42	4.40±0.24	81.40±0.55	74.79±0.88	4.18±2.03	3.47±0.23	20.72±0.82
	KFLLA	85.59±0.50	4.71±0.30	80.82±0.48	75.45±1.11	5.84±2.21	6.93±0.00	21.86±1.19
	SV-DKL	85.69±0.66	5.58±0.79	78.54±2.20	75.32±0.84	8.52±1.57	4.49±0.65	23.10±1.66
	SGPA	85.39±0.36	4.96±0.49	80.04±1.14	76.44±0.96	6.04±1.71	3.96±0.50	22.19±1.06
	Transformer	85.59±0.50	4.73±0.32	80.75±0.55	75.45±1.10	6.96±2.05	3.95±0.44	22.28±1.26
	DIRECTOR	**86.07**±0.61	**4.57**±0.39	**80.84**±0.67	**74.24**±0.87	**5.40**±1.90	**3.60**±0.34	**21.08**±1.32
	KEP-1/5	85.76±0.71	4.54±0.42	81.02±0.70	74.87±0.87	5.51±2.94	3.79±0.51	21.62±1.42
	DIRECTOR	**87.13**±0.19	**4.07**±0.30	**81.55**±0.62	**73.35**±0.14	**3.16**±2.54	**3.24**±0.31	**19.31**±0.97
	KEP-2/5	86.52±0.72	4.18±0.37	81.54±0.54	73.82±1.68	5.72±1.20	3.58±0.29	20.51±1.16
	DIRECTOR	**86.95**±0.19	**4.05**±0.30	**81.64**±0.85	**73.46**±1.34	**3.46**±1.40	**3.23**±0.13	**19.42**±0.49
	KEP-5/5	84.57±0.81	5.48±0.60	79.32±1.25	77.03±1.48	7.83±3.28	5.17±2.13	24.23±2.56
	DIRECTOR	**85.74**±0.34	**4.58**±0.23	**80.95**±0.42	**75.08**±1.15	**2.33**±1.54	**3.36**±0.12	**20.70**±0.65
CoLA	TS	29.92±1.17	20.84±1.23	64.31±1.44	89.93±2.95	23.22±2.99	11.04±1.91	51.70±3.42
	MCD	30.04±1.02	20.66±1.11	64.53±1.00	89.51±1.35	24.96±1.79	17.83±3.61	53.52±2.59
	KFLLA	29.89±1.14	20.82±1.26	64.22±1.46	89.87±3.24	24.36±2.25	12.16±1.49	52.80±2.85
	SV-DKL	30.07±1.41	22.76±2.28	61.98±3.09	**89.00**±2.55	25.71±1.60	17.96±3.26	54.40±2.13
	SGPA	31.53±2.05	20.44±2.60	64.34±1.95	90.79±0.87	26.22±1.51	28.65±7.23	54.08±2.44
	Transformer	29.92±1.17	20.80±1.21	64.22±1.46	90.01±2.84	26.44±1.90	19.66±4.18	55.09±2.68
	DIRECTOR	**31.85**±2.46	**19.74**±1.62	**64.52**±1.71	**89.52**±3.61	**23.94**±0.49	**14.29**±3.01	**50.82**±1.19
	KEP-1/5	30.86±2.03	19.86±2.04	**65.18**±1.83	**89.10**±2.60	24.98±2.08	16.28±4.35	52.86±2.88
	DIRECTOR	**31.84**±1.88	**19.42**±1.83	**65.54**±1.78	90.37±0.89	**13.77**±6.58	**8.26**±3.75	**43.54**±3.65
	KEP-2/5	30.73±2.29	20.36±1.40	64.03±1.21	90.23±1.86	21.78±6.62	15.52±7.74	50.14±5.92
	DIRECTOR	**33.03**±1.36	**19.06**±3.05	**64.78**±2.31	**89.79**±2.65	**11.45**±4.78	**6.72**±0.98	**41.64**±3.58
	KEP-5/5	29.28±1.21	20.82±1.98	64.47±0.90	89.65±1.04	18.95±5.66	11.83±7.96	48.65±5.84
	DIRECTOR	**32.05**±1.56	**18.69**±1.39	**64.71**±1.37	**89.53**±1.67	**12.88**±5.74	**7.68**±1.70	**42.20**±2.50

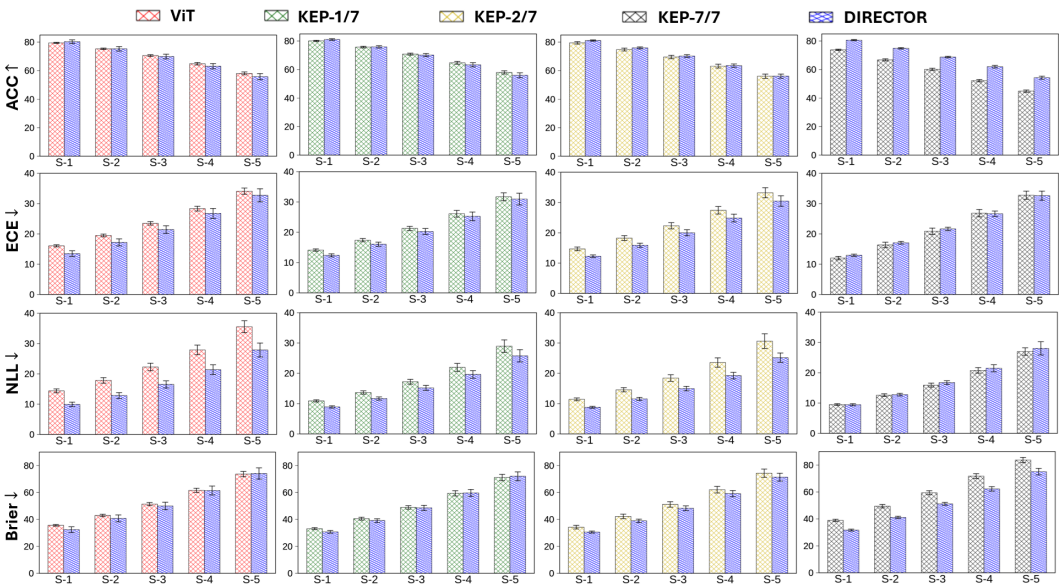


Figure 3: Calibration comparison of pre-trained models with their corresponding diffusion-based reconfigured produced by DIRECTOR on CIFAR-10-C over 5 severity levels of corruption. The notation `S-k` represents the severity level k. DIRECTOR achieves competitive accuracy and outperforms pre-trained models in most calibration metrics.

Table 19: Performance comparison on CIFAR-10-C, with 15 corruptions across five severity levels over five trials.

Method	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
MCD	69.73±0.36	14.46±0.22	79.11±0.11	76.96±0.20	19.58±0.23	18.48±0.67	48.70±0.47
KFLLA	69.64±0.43	14.46±0.41	79.27±0.23	76.34±0.33	17.46±1.18	12.75±0.80	47.18±1.06
SVDKL	68.88±0.36	15.43±0.56	78.56±0.50	77.22±0.55	22.67±1.18	21.02±2.57	52.29±1.35
SGPA	57.73±1.25	24.82±1.23	74.81±0.65	80.92±0.54	12.47±2.11	13.60±0.48	57.91±1.39
ViT	**69.67**±0.34	**14.66**±0.27	78.92±0.16	77.74±0.28	24.30±0.31	23.59±1.00	53.07±0.59
DIRECTOR	68.89±1.44	15.31±1.30	**79.17**±0.86	**77.10**±1.00	**22.32**±1.09	**17.71**±1.02	**51.77**±2.39
KEP-1/7	**69.87**±0.45	**14.30**±0.50	79.49±0.38	76.95±0.43	22.12±0.47	18.54±0.63	50.65±0.90
DIRECTOR	69.29±0.66	15.00±0.69	**79.69**±0.27	**76.38**±0.40	**20.98**±0.65	**16.23**±0.60	**50.07**±1.20
KEP-2/7	68.63±0.75	15.46±0.66	78.90±0.37	77.57±0.42	23.18±0.63	19.72±0.76	52.82±1.30
DIRECTOR	**69.41**±0.55	**14.95**±0.60	**79.73**±0.42	**76.29**±0.73	**20.71**±0.63	**15.94**±0.44	**49.71**±1.10
KEP-7/7	59.57±0.30	23.77±0.40	75.56±0.30	80.47±0.27	**21.78**±0.59	**17.17**±0.39	60.67±0.72
DIRECTOR	**68.12**±0.26	**16.45**±0.28	**79.08**±0.22	**76.73**±0.29	22.19±0.19	17.70±0.17	**52.27**±0.35

performance. Conversely, assigning too much weight to λ_{NLL}, as in $(0.05, 0.05, 0.9)$, prioritizes accuracy at the expense of calibration, leading to more confident yet less well-calibrated predictions. Intermediate configurations, such as $(0.5, 0.2, 0.3)$, provide a balanced trade-off, achieving high predictive performance while significantly improving calibration. Based on these insights, we adopt $(0.5, 0.2, 0.3)$ consistently for all diffusion-based reconfigured KEP produced by DIRECTOR, as it offers the most reliable combination of accuracy and calibrated uncertainty.

We also study the effect of varying the weights for the mean matching term (λ_{mean}) and the performance-aware loss (λ_{NLL}) while setting $\lambda_{\text{Chol}} = 0$ during diffusion-based reconfiguring ViT with DIRECTOR on CIFAR-10 (Table 22). When λ_{mean} is excluded (e.g., $(0.0, 1.0)$), the NLL term dominates, which may slightly improve calibration but leads to reduced predictive performance. In contrast, including both mean matching ($\lambda_{\text{mean}} \neq 0$) and performance-aware loss consistently enhances accuracy and uncertainty calibration. Notably, the configuration $(0.8, 0.2)$ achieves the best overall balance, with the highest accuracy (87.16) and lowest ECE (8.54). Consequently, we adopt $(0.8, 0.2)$ for all diffusion-based reconfigured ViT produced by DIRECTOR.

Table 20: Performance comparison on CoLA OOD over five trials.

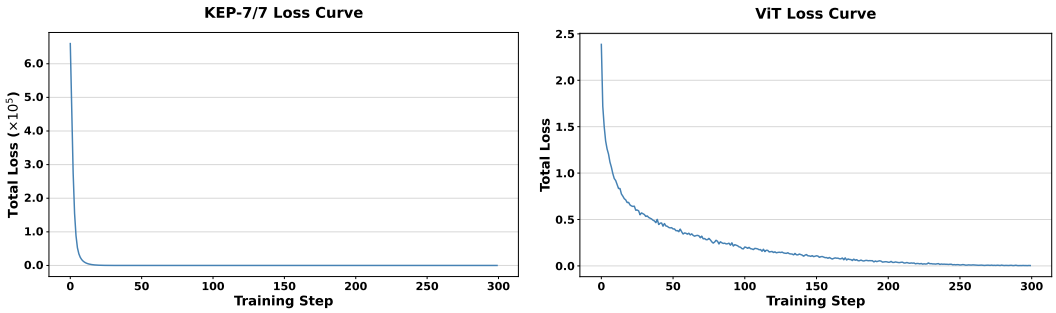
Method	MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
MC Dropout	18.54±4.14	25.85±1.03	63.31±2.18	90.17±2.95	32.02±2.56	23.84±5.19	65.50±4.46
KFLLA	18.43±3.55	25.89±0.99	63.31±1.65	90.50±3.25	29.94±2.86	14.64±1.87	62.72±4.37
SVDKL	19.32±3.57	27.58±2.51	60.65±1.41	89.97±2.67	30.97±3.10	21.23±3.67	64.07±4.62
SGPA	19.34±6.23	27.48±2.55	61.68±2.29	90.14±2.30	31.15±1.66	35.18±8.59	63.62±3.65
Transformer	18.43±3.55	25.85±1.00	**63.38**±1.69	90.39±3.27	31.99±2.70	23.82±5.16	65.50±4.32
DIRECTOR	**23.06**±4.69	**25.61**±2.30	61.32±3.21	**88.23**±3.89	**28.72**±1.87	**17.18**±3.75	**59.91**±2.20
KEP-1/5	19.44±1.94	25.21±1.53	**63.65**±3.19	**87.35**±2.43	30.33±1.48	19.67±4.82	61.97±2.25
DIRECTOR	**22.10**±5.49	**24.91**±1.80	62.23±2.98	91.55±3.19	**17.50**±7.52	**9.34**±4.50	**49.92**±5.80
KEP-2/5	**20.38**±5.22	24.39±2.05	63.60±1.71	90.75±2.62	26.04±7.51	17.94±9.12	57.77±8.04
DIRECTOR	20.11±2.48	**23.23**±2.36	**63.62**±0.91	**89.81**±1.37	**15.67**±5.90	**7.64**±1.45	**48.23**±5.01
KEP-5/5	**21.14**±3.48	24.06±2.27	63.33±1.36	90.46±2.20	23.27±6.61	14.25±10.66	55.12±7.78
DIRECTOR	20.20±5.46	**21.97**±1.72	**65.12**±1.63	**90.17**±3.89	**17.49**±5.93	**8.61**±1.99	**48.55**±3.67

Table 21: Ablation on loss weighting on CoLA when diffusion-based reconfiguring KEP-5/5 with DIRECTOR. We report mean ± std across 5 seeds. The configuration $(\lambda_{\text{mean}}, \lambda_{\text{Chol}}, \lambda_{\text{NLL}}) = (0.5, 0.2, 0.3)$ (gray) is chosen as it balances predictive performance (MCC) with uncertainty calibration. This setting is applied consistently for all diffusion-based reconfigured KEP (Chen et al., 2024b) produced by DIRECTOR .

Method	λ_{mean}	λ_{Chol}	λ_{NLL}	MCC ↑	ECE ↓	NLL ↓
Transformers	–	–	–	29.92±1.17	26.44±1.90	19.66 ± 4.18
DIRECTOR	0.9	0.05	0.05	29.98±3.45	5.96±4.60	5.86±0.49
	0.6	0.1	0.3	33.18±1.53	18.36±2.29	10.09±2.38
	0.5	0.3	0.2	32.26±2.87	15.27±9.06	9.51±2.70
	0.5	0.25	0.25	32.96±1.85	12.27±8.57	8.23±2.71
	0.5	0.2	0.3	31.81±2.30	12.65±6.04	7.69±1.70
	0.3	0.2	0.5	31.86±2.79	18.64±8.43	13.77±7.32
	0.3	0.1	0.6	30.21±2.32	16.50±8.60	10.74±5.80
	0.25	0.25	0.5	33.38±2.08	22.26±3.72	15.10±4.85
	0.05	0.05	0.9	30.74±1.44	15.50±6.34	9.52±4.12

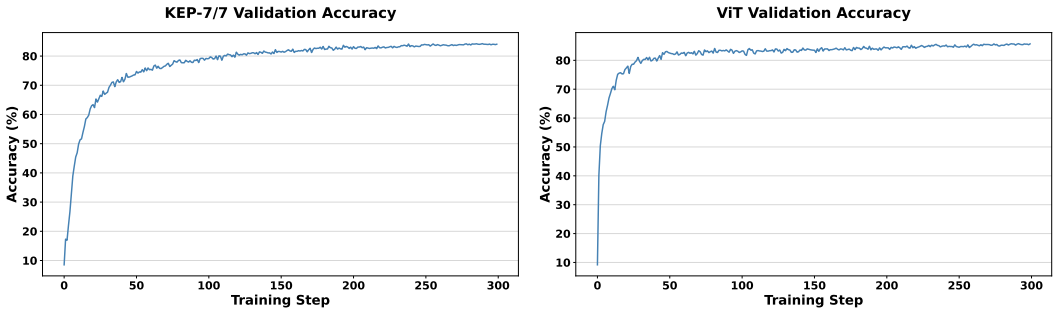
Table 22: Ablation on loss weighting on CIFAR-10 (single run) when reconfiguring ViT with DIRECTOR. We report test accuracy (ACC), Expected Calibration Error (ECE), and Negative Log-Likelihood (NLL). The configuration $(\lambda_{\text{mean}}, \lambda_{\text{NLL}}) = (0.8, 0.2)$ (gray) is selected as it achieves the best balance between accuracy and calibration, and is used for all diffusion-based reconfigured ViT produced by DIRECTOR .

Method	λ_{mean}	λ_{NLL}	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
ViT	–	–	84.22	3.90	86.62	65.34	12.39	10.67	27.52
DIRECTOR	0.0	1.0	84.03	4.01	85.97	67.81	12.12	9.69	27.49
	0.1	0.9	85.24	3.26	87.54	62.87	10.97	8.51	24.99
	0.2	0.8	85.67	3.03	88.16	61.83	10.52	7.75	24.11
	0.3	0.7	86.08	3.06	87.57	59.91	10.24	7.86	23.44
	0.4	0.6	86.05	2.91	88.51	62.22	10.25	7.61	23.45
	0.5	0.5	86.18	2.86	88.21	63.89	9.97	7.41	23.45
	0.6	0.4	86.60	2.83	87.81	64.93	9.50	6.98	22.58
	0.7	0.3	86.66	2.73	88.39	61.02	9.18	6.74	22.05
	0.8	0.2	**87.16**	**2.52**	**88.92**	**57.94**	**8.54**	**6.06**	**20.99**
	0.9	0.1	85.09	3.54	86.73	64.19	10.15	7.37	24.79



(a) KEP-7/7 Loss Curve

(b) ViT Loss Curve



(c) KEP-7/7 Validation Accuracy

(d) ViT Validation Accuracy

Figure 4: Training curves comparison between KEP-7/7 and ViT models. (a) and (b) show the loss curves, while (c) and (d) show the validation accuracy over training steps.

A.6.13 Training Configuration and Learning Dynamics

Pre-trained transformer-based models. We reproduce the pre-trained ViT and KEP-SVGP models following the experimental settings in Chen et al. (2024b), as their weights are not publicly available. To ensure that the reproduced models converge properly, we report training loss and validation accuracy over epochs for ViT and KEP-7/7 on CIFAR-10 in Figure 4. The curves indicate stable convergence for both architectures, providing a reliable starting point for subsequent reconfiguration.

DIRECTOR. Building on these pre-trained models, we reconfigure each using a DiT-based unified transition model that takes (X_t, t) as input and outputs the feature representation at diffusion step t. This unified spatiotemporal transition model is implemented using a single-block DiT, which conditions on the diffusion timestep via adaptive LayerNorm-zero (AdaLN-Zero). We adopt the default DiT configuration while adjusting the model dimension d_{model} to match the corresponding pre-trained backbone, ensuring compatibility. For reproducibility, all training hyperparameters for the unified transition model are summarized in Table 23.

A.7 Broader Impact

This work advances the reliability and interpretability of large pre-trained Transformer models by enabling principled uncertainty calibration, a critical requirement for deploying AI in high stakes domains such as healthcare, autonomous systems, and scientific discovery. By introducing a scalable framework for embedding probabilistic reasoning within existing architectures, our approach promotes safer and more accountable AI systems. Additionally, it opens new opportunities for research in uncertainty-aware foundation models and supports the responsible use of AI in socially impactful applications.

Table 23: Hyperparameters for unified transition model training.

Hyperparameters	Settings
d_{model}	384 (CV), 256 (CoLA), 128 (IMDB)
Number of attention heads	12 (CV), 8 (CoLA) , 4 (IMDB)
Depth of DiT	1
Epochs	100 (CV), 50 (NLP)
Batch size	128 (CV), 32 (NLP)
Dropout	0.1
Optimizer	Adam
Learning rate	1e-3 (CV), 5e-3 (NLP)
Adam β	(0.9, 0.999)
Weight decay	1e-5
Scheduler	CosineAnnealingLR
Cosine cycle epochs	50
Minimal learning rate	1e-5
Warmup Epochs	5

A.8 LIMITATIONS

While our approach enables principled uncertainty propagation in pre-trained Transformers, it introduces additional computational overhead by absorbing the pre-trained probability path into the propagation model. As the propagation model must be configured with a size compatible with that of the pre-trained model, the computational cost grows proportionally with model scale. Consequently, large-scale or very deep architectures in real-world settings require further investigation to validate the efficacy of the proposed method, which we leave for future work due to computational constraints. In addition, although we demonstrate strong empirical performance across vision and language tasks, further evaluation is needed in domains with sparse supervision or non-stationary data distributions.