AdUE: Improving uncertainty estimation head for LoRA adapters in LLMs

Anonymous ACL submission

Abstract

Uncertainty estimation is a key challenge when adapting pre-trained language models to downstream classification tasks. It is crucial to identify instances where the model might make a mistake. Our approach, AdUE[1] is a lightweight uncertainty estimation method based on two key components. First, we initialize a new uncertainty estimation head using the original classification head's weights, so that its initial outputs (entropy scores) match the original model's entropy-based uncertainty. Second, we fine-tune this head with two regularizers—an output consistency regularization and an L2-SP weight anchoring—to improve robustness during training. Experiments across five NLP classification datasets and four language models (RoBERTa, ELECTRA, LLaMA-2, and Qwen) show that AdUE consistently outperforms strong baselines, including vanilla entropy, softmax response, linear, and attention probing. Our method requires a lightweight additional head to a base model, while significantly improving the ROC–AUC for distinguishing correct vs. incorrect predictions, compared to strong baselines.

1 Introduction

Large language models are foundational in NLP, achieving strong transfer across tasks. For classification, *parameter-efficient fine-tuning*—adapters (Houlsby et al., 2019) and LoRA (Hu et al., 2022) have emerged as scalable alternatives to full model fine-tuning. These approaches update only a small subset of parameters, retaining most of the pre-trained backbone. This enables efficient adaptation with greatly reduced computation and storage overhead.

Adapter-based models can solve a wide range of tasks, including risk-sensitive applications (e.g., medical triage, toxic content filtering, automated moderation). A principled estimate of predictive *uncertainty*—how confident the model is in its outputs—is therefore essential. Common approaches include entropy of the predictive distribution, Softmax Response (Geifman and El-Yaniv, 2017), Mahalanobis distance (Lee et al., 2018), and deep ensembles (Lakshminarayanan et al., 2017); however, these methods involve trade-offs among computational cost, scalability, and generalization.

We revisit predictive entropy as a lightweight yet effective uncertainty measure and propose AdUE, a fine-tuning strategy to further improve entropy-based uncertainty estimates. Our **AdUE** operates on a frozen, task-tuned adapter backbone (the LoRAadapted model) and introduces a new uncertainty head initialized to reproduce the model's vanilla entropy scores. During training, AdUE optimizes a multi-task loss with three components: (i) a binary cross-entropy term to predict downstream errors, (ii) a consistency regularization term to keep the new head's outputs close to the original entropy scores, and (iii) an L2-SP anchoring term to keep the head's weights near their initialization (Xuhong et al., 2018). The intuition is that, since we start from a strong baseline (entropy), the fine-tuned uncertainty head should not deviate too far from the original in either weight space or output space. Moreover, classification heads often contain more parameters than there are labeled training examples; thus, these regularizers are crucial to prevent overfitting.

Our main contributions are as follows:

- We propose AdUE — a lightweight post-hoc fine-tuning method based on probability entropy uncertainty estimation in adapter-based LLMs. Figure 1 illustrates the approach.
- We introduce a multi-component loss function that balances error prediction accuracy, output consistency with the original entropy, and parameter anchoring for improved robustness.

[1] https://anonymous.4open.science/r/AdUE-1824/

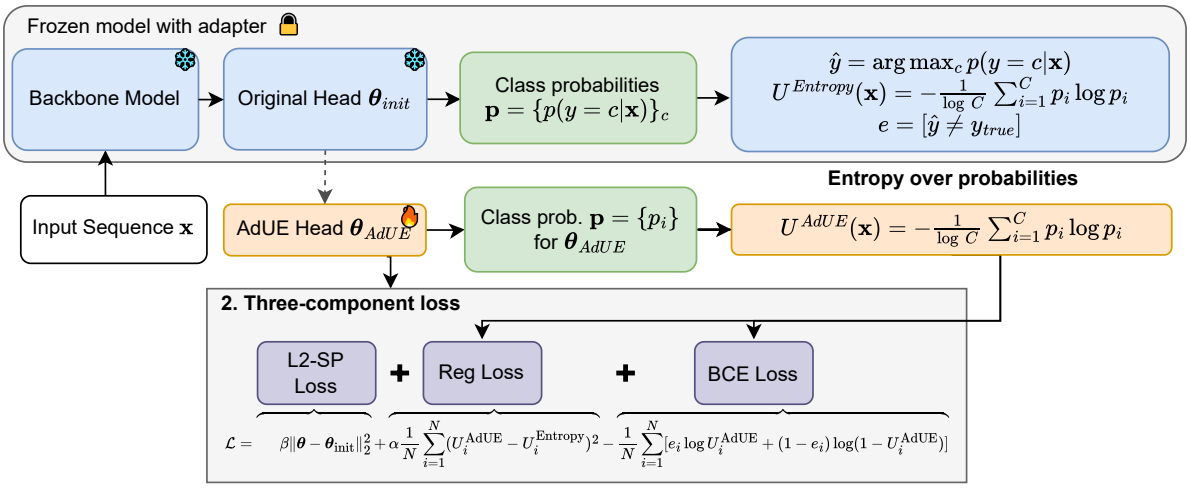

Figure 1: AdUE head training scheme. We initialize the new uncertainty head with the original classifier's weights θ_{init}, use its entropy output as an initial uncertainty signal, and then fine-tune the head with the three-term loss (binary cross-entropy, output consistency regularization, and L2-SP anchoring), while keeping the rest of the model (backbone+adapter) frozen.

- We evaluate AdUE on five NLP classification benchmarks and four LoRA-tuned LLMs, demonstrating consistent improvements in uncertainty estimation over strong baseline methods, measured by ROC—AUC.

2 Background Uncertainty Estimation Methods

Let us consider a multi-class classification problem with the model goal being the prediction of the true class. For a classification model $y(\mathbf{x})$ with classes $c \in \mathcal{C}$, the model produces a vector of class probabilities estimates $\{p(c|\mathbf{x})\}_{c\in C}\}$ with $\mathbf{x}$ being a model input. We aim to provide an uncertainty score $U(\mathbf{x})$ that reflects the possibility of an error for an example $\mathbf{x}$ for a model. The study considers methods that require low computational cost and are efficient if compared to a single LLM run.

Within the problem statement, a model also produces a latent representation $\mathbf{h} = h(\mathbf{x}) \in \mathbb{R}^d$ for an input instance $\mathbf{x}$. The vector $\mathbf{h}$ is typically extracted from the penultimate layer of a neural network, while $\mathbf{x}$ for LLMs is a natural language text.

2.1 Probability-based uncertainty estimates

Softmax Response (SR) (Geifman and El-Yaniv, 2017) measures uncertainty via the complement of the maximum class probability:

$$U^{\text{SR}}(\mathbf{x}) = 1 - \max_{c\in\mathcal{C}} p(c \mid \mathbf{x}), \qquad (1)$$

where $p(c \mid \mathbf{x})$ are the softmax probabilities.

Entropy quantifies uncertainty of the predictive distribution:

$$U^{\text{Entropy}}(\mathbf{x}) = -\sum_{c=1}^{C} p(c \mid \mathbf{x}) \log p(c \mid \mathbf{x}). \qquad (2)$$

For a binary classification, entropy reduces to SR. For a multiclass setting, it exploits the full distribution and provides superior results.

3 Methodology

We use LoRA adapters for efficient, robust fine-tuning of large LMs. After task fine-tuning, we freeze the backbone and adapter, then attach an uncertainty head to the penultimate layer. Initialized from the original classifier (the logits layer used for entropy), this head is fine-tuned—with output/parameter regularization—to predict the probability of misclassification.

3.1 AdUE. Probability-entropy fine-tuning

Probability-based uncertainty scores Softmax Response (1) and predictive entropy (2) perform well without extra training (Vazhentsev et al., 2023; Holm et al., 2023). We introduce an AdUE head (a duplicate of the trained classifier layer) which initially produces the exact same entropy scores as the original model. By further fine-tuning of AdUE head, we are able to produce more suitable scores that predict, for each sample, the probability that the model's response is incorrect.

Let $\text{AdUE}(h(\mathbf{x}))$ denote the logits produced by the new uncertainty head for input $\mathbf{x}$. We obtain

	LLaMA					Qwen					Avg.
	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	Rank
MD	60.5±3.9	56.3±2.74	59.7±9.35	51.3±0.98	56.4±2.05	56.7±2.45	54.0±0.69	54.6±5.5	52.2±1.14	58.1±5.17	10.7
MDR	67.7±5.9	80.7±1.16	79.1±2.49	53.3±0.57	67.5±2.56	64.9±4.63	81.1±0.62	78.6±4.24	54.5±0.77	66.5±1.71	5.9
MDM	60.5±3.88	55.9±2.7	59.6±9.35	51.3±0.98	56.3±2.04	56.6±2.43	53.5±0.71	54.6±5.43	52.2±1.14	58.1±5.17	11.7
RDE	61.3±5.73	45.9±5.01	62.8±11.82	48.9±2.4	53.2±2.36	47.4±3.41	47.8±3.68	59.0±12.12	51.4±2.91	51.3±4.84	12.7
Linear probe last	65.8±5.98	75.5±1.47	87.8±1.51	53.1±2.5	56.8±6.15	72.0±8.67	71.5±1.82	86.2±2.02	55.1±2.18	61.4±4.55	5.8
Linear probe middle	59.0±2.7	69.4±1.71	88.0±0.69	52.8±2.3	60.7±2.07	63.3±6.93	61.9±1.36	82.8±2.51	54.2±2.09	57.0±2.5	8.0
Linear probe begin	55.6±7.57	59.5±1.87	72.5±3.09	53.0±1.94	54.7±2.89	62.2±6.2	58.3±1.22	64.3±3.0	51.6±1.62	56.2±3.4	10.4
Attention pooling last	64.5±7.73	78.7±1.26	87.5±1.5	51.0±1.58	53.7±5.05	64.0±16.21	70.6±1.57	85.2±3.29	52.8±2.12	56.8±6.41	8.3
Attention pooling middle	57.0±4.63	75.6±1.25	87.2±1.46	51.5±0.86	54.3±8.01	60.4±8.28	68.0±1.7	77.4±9.14	51.9±1.22	55.1±7.07	9.9
Attention pooling begin	61.2±2.55	69.1±5.33	71.7±11.9	52.5±1.44	53.2±5.29	60.0±5.14	57.4±5.93	56.0±9.08	51.0±0.87	54.4±4.89	11.4
SR	78.2±0.89	86.3±0.45	88.5±1.53	59.5±2.38	76.7±1.57	78.4±1.94	85.6±0.46	86.1±2.12	60.7±1.09	72.2±2.78	3.6
Entropy	78.5±1.02	86.4±0.41	88.5±1.53	58.3±1.93	77.3±1.6	78.4±1.96	85.9±0.45	86.1±2.12	59.5±1.19	74.1±2.92	3.2
SRFT (ours)	77.9±0.77	**88.6±0.53**	88.7±1.64	62.7±1.31	76.8±2.49	79.7±1.6	87.8±0.23	**86.6±1.94**	62.8±1.47	73.2±2.36	2.2
AdUE (ours)	**78.7±0.48**	88.6±0.42	**88.9±1.62**	**62.8±1.22**	**77.7±2.46**	**80.5±1.4**	**88.0±0.29**	86.5±1.57	**62.8±1.38**	**75.0±1.76**	**1.2**

	Electra					Roberta					Avg.
	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	Rank
MD	**80.9±1.08**	74.1±0.78	**88.1±0.9**	55.6±0.96	73.9±1.44	73.6±0.38	79.4±0.41	**82.7±2.29**	55.6±1.79	65.0±9.12	5.5
MDR	69.8±2.65	83.7±0.62	79.0±3.86	57.3±1.39	67.8±1.48	65.0±1.7	84.6±0.68	72.7±2.57	58.0±1.09	69.1±2.25	6.5
MDM	80.9±1.06	73.2±0.74	88.1±0.91	55.6±0.96	73.9±1.44	73.6±0.38	79.0±0.41	82.7±2.29	55.5±1.79	65.0±9.14	6.6
RDE	77.9±4.74	65.4±0.91	87.1±1.9	56.4±0.84	74.2±1.14	73.1±1.33	76.8±0.74	81.5±2.24	56.1±1.39	64.8±8.96	7.2
Linear probe last	57.3±6.82	78.2±0.47	80.5±17.69	55.8±1.68	60.4±9.63	62.6±6.17	75.3±1.05	62.2±10.28	57.1±2.5	61.5±10.52	9.2
Linear probe middle	60.5±2.12	73.4±1.23	61.5±3.89	51.9±1.65	52.1±4.41	62.5±4.05	68.9±1.64	55.8±4.29	53.5±1.67	56.2±3.03	12.0
Linear probe begin	60.8±0.98	70.7±1.04	58.2±4.73	51.8±0.53	50.7±4.49	60.7±2.84	59.9±0.83	52.8±3.11	52.4±1.19	55.2±1.31	13.4
Attention pooling last	65.2±2.06	78.0±0.56	86.4±4.04	55.9±1.86	64.5±7.61	61.6±4.58	75.5±1.11	63.6±10.84	56.2±2.13	65.8±6.65	8.3
Attention pooling middle	56.5±2.56	75.4±0.67	61.4±6.28	53.5±1.34	58.5±4.09	61.9±6.06	71.9±1.77	62.6±3.38	55.3±2.53	59.8±5.41	11.0
Attention pooling begin	54.4±3.02	74.4±0.68	60.7±3.71	52.3±1.77	57.8±2.97	55.7±3.34	60.9±1.75	57.3±5.29	53.9±1.59	61.7±4.19	12.1
SR	79.2±1.5	83.4±0.63	86.4±2.04	60.8±1.47	74.4±1.28	75.0±2.39	84.7±0.19	80.9±2.04	60.9±1.64	74.2±0.72	4.4
Entropy	79.1±1.5	83.7±0.64	86.4±2.05	59.8±1.35	74.5±1.38	75.0±2.36	84.9±0.21	80.9±2.01	59.4±0.87	**74.7±0.31**	4.1
SRFT (ours)	80.0±0.92	84.7±0.47	87.0±2.67	62.2±0.89	74.9±0.94	75.1±3.1	85.9±0.26	81.6±2.09	**62.1±1.11**	74.1±0.86	2.7
AdUE (ours)	80.4±0.89	**85.0±0.46**	87.1±2.79	**62.5±1.13**	**74.9±0.39**	**75.5±2.84**	**86.0±0.22**	81.5±1.98	61.6±1.27	74.5±0.46	**2.0**

Table 1: ROC–AUC (mean ± std over five independent seeds) for misclassification detection across five datasets and four architectures. For each seed, we re-train the LoRA-tuned base model from scratch, then apply and evaluate all UE methods (baselines and AdUE) on the resulting model. AdUE consistently improves or matches the best performance on most datasets and models.

class probabilities via a softmax:

$$p(c \mid \mathbf{x}) = \mathrm{softmax}(\mathrm{AdUE}(h(\mathbf{x})))_c, \quad (3)$$

for each class c. We initialize the head's parameters as $\theta_{\mathrm{AdUE}} \leftarrow \theta_{\mathrm{init}}$ (the original classifier's weights) so that initially $p(c|x)$ matches the original classifier's outputs

We define the AdUE uncertainty score as the normalized entropy of this predictive distribution:

$$U_{\mathrm{AdUE}}(\mathbf{x}) = -\frac{1}{\log C} \sum_{c=1}^{C} p(c \mid \mathbf{x}) \log p(c \mid \mathbf{x}), \quad (4)$$

which is bounded between 0 and 1. By normalizing by $\log C$, we ensure that $U_{\mathrm{AdUE}}(\mathbf{x}) = 1$ corresponds to maximum uncertainty (uniform distribution) and 0 corresponds to minimum uncertainty (a confident one-class output).

3.2 Training Objective

The smooth head with parameters $\boldsymbol{\theta}_{\mathbf{AdUE}}$ is trained using a multi-task objective that combines three components, which takes into account model error $e_i = [\hat{y} \neq y_{true}]$, uncertainty estimates $U^{\mathrm{Entropy}}(\mathbf{x})$, and parameters $\boldsymbol{\theta}_{\mathrm{init}}$ from the original head: The first is a classic binary cross-entropy loss:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \Big[e_i \log U_i^{\mathrm{AdUE}} + (1 - e_i) \log(1 - U_i^{\mathrm{AdUE}}) \Big].$$

The next term is a regularization loss:

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{N} \sum_{i=1}^{N} \Big(U_i^{\mathrm{AdUE}}(\mathbf{x}_i) - U^{\mathrm{Entropy}}(\mathbf{x}_i) \Big)^2. \quad (5)$$

This encourages the new uncertainty scores to stay close to the original probability entropy confidence on average, preventing drastic shifts.

Finally, we add L2-SP loss adopted from the transfer learning domain (Xuhong et al., 2018), which keeps the fine-tuned weights near their initialization, avoiding forgetting and overfitting:

$$\mathcal{L}_{\mathrm{L2SP}} = \|\boldsymbol{\theta}_{\mathrm{AdUE}} - \boldsymbol{\theta}_{\mathrm{init}}\|_2^2. \quad (6)$$

Our full training objective is a weighted sum of these components:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \alpha \mathcal{L}_{\text{reg}} + \beta \mathcal{L}_{\text{L2SP}}. \quad (7)$$

4 Experiments

4.1 Evaluation protocol

We follow an established protocol (Fadeeva et al., 2024) and evaluate on five text-classification datasets (SST-2, SST-5, CoLA, 20 Newsgroups, ToxiGen). See details in Appendix E. We cast uncertainty quantification as predicting whether to *trust* a model's prediction and report ROC–AUC, which measures how well the scores separate correct from incorrect predictions (Wang et al., 2024). Then the validation assesses uncertainty via ROC-AUC for the error prediction problem.

Models. We evaluate AdUE on four pre-trained base models: two smaller encoders, roberta-base Liu et al. (2019) (125M parameters; trained with Masked Language Modeling) and electra-base Clark et al. (2020) (110M; Replaced Token Detection), as wella as two 7-billion-parameter decoder LLMs, Qwen2.5-7B (Yang et al., 2024) and LLaMA-2-7B (Touvron et al., 2023).

Baselines. A pool of baselines includes 8 strong approaches with 12 of their variations considered. A nuanced discussion of the prior work, as well as the definition of baselines, appears in Appendices B and C. Additionally, we compare our AdUE approach with a similar method based on fine-tuning of a smoothed max probability head, specifically *Softmax response fine-tune (SRFT)*.

Technical details. For each base model, we train five LoRA adapters with distinct random seeds (which affect data shuffling and parameter initialization). On each trained instance, we then train AdUE and all baselines, as well as optimizing hyperparameters. Further details on hyperparameters are in Appendix D.

4.2 Main results

The main results are presented in Table 1. AdUE achieves the highest ROC–AUC in 17 out of 20 dataset–model combinations, for both encoder and decoder-based models. The second-best approach is also ours, SRFT, which fine-tunes the maximum probability for the SR head. Entropy is the second-best baseline, despite numerous more complex methods being proposed. We hypothesize that this outcome stems from the limited size of the fine-tuning datasets, which may prevent more complex uncertainty methods from realizing their full potential. In low-data settings, the model's own entropy (a single-model, single-forward-pass method) remains strong, whereas methods that require additional training probes or fitting distance metrics overfit or lack sufficient data volumes to excel. Notably, on SST-2 (the largest fine-tuning set among our benchmarks), some advanced baselines close the gap with entropy, consistent with our hypothesis.

4.3 Sensitivity study

As shown in Table 2, using the full classifier initialization with all loss terms AdUE yields the best performance. Any deviation — using a random head or dropping a regularization term — led to a drop in ROC-–AUC and a worse average rank. This confirms that both the initialization and each loss component are required for optimal results. Further details and results are provided in the Appendix G.

5 Conclusion

We introduced **AdUE**, a lightweight post-hoc uncertainty head for LoRA-tuned LLMs. AdUE starts from the model's own predictive entropy and refines it to better quantify uncertainty. By copying the trained classifier head with probability entropy and optimizing a three-term objective (binary cross-entropy, output consistency L_{reg}, and parameter anchoring $L_{L2\text{-}SP}$), AdUE consistently improves the separation between correct and incorrect predictions, outperforming both classical confidence baselines, probing and more robust distance-based baselines in nearly all evaluations. AdUE adds only 0.1% additional parameters and incurs negligible computation overhead during inference with one extra forward pass of a small layer. Ablations confirm that each loss component is necessary: using a random initialization or dropping either regularizer lowers quality, whereas the full objective yields controlled adaptation and reduced seed-to-seed variance. Looking ahead, combining AdUE with post-hoc calibration techniques or incorporating uncertainty awareness during training are promising directions to further improve reliability.

6 Limitations

While our method improves uncertainty estimation, several limitations remain. On the SST-2 dataset, where encoders were originally trained on the data-rich SST-2 dataset (x5-x10 times more training samples), AdUE underperformed relative to some baselines. This finding suggests that the effectiveness of our approach may depend significantly on the characteristics of the training set, and traditional methods may be more effective in data-rich scenarios.

Additionally, we recognize that the evaluation scope could be expanded. First, our experiments focus exclusively on adapter-based fine-tuning, specifically LoRA, and do not explore other parameter-efficient methods such as prompt or prefix tuning, which might exhibit different uncertainty characteristics. Second, our assessment focuses solely on classification tasks. The proposed entropy head has not yet been evaluated in generation tasks or other contexts, such as sequence labeling, where uncertainty dynamics differ, and factors like beam diversity, exposure bias, or semantic identification (Farquhar et al., 2024) become relevant. This evaluation presents additional challenges, as the generative entropy-based head would contain many more parameters, making training without overfitting even more difficult. Lastly, our primary evaluation metric, uncertainty–error AUC, does not directly measure calibration metrics such as Expected Calibration Error (ECE). Thus, one can include a calibration-aware loss term in our full loss function to address the additional calibration objective (Shen et al., 2024). Future research should address these limitations, extending the analysis to additional architectures, tasks, and evaluation frameworks.

References

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *EMNLP*.

Pierre Colombo, Maxime Darrin, and Pablo Panitainada. 2023. Rainproof: An umbrella to shield text generators from out-of-distribution data. In *EMNLP*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Andreas Nugaard Holm, Dustin Wright, and Isabelle Augenstein. 2023. Revisiting softmax for uncertainty approximation in text classification. *Information*, 14(7).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

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

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. 2021. A simple fix to mahalanobis distance for improving near-ood detection. *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory Wornell, and Soumya Ghosh. 2024. Thermometer: Towards universal calibration for large language models. In *International Conference on Machine Learning*.

CH-Wang Sky, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4401–4420.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Junya Takayama and Yuki Arase. 2019. Relevant and informative response generation using pointwise mutual information. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 133–138.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252, Dublin, Ireland. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681.

Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Hengtao Shen, and Xiaofeng Zhu. 2024. Conu: Conformal uncertainty in large language models with correctness coverage guarantees. *Preprint*, arXiv:2407.00499.

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

Li Xuhong, Yves Grandvalet, and Franck Davoine. 2018. Explicit inductive bias for transfer learning with convolutional networks. In *International conference on machine learning*, pages 2825–2834. PMLR.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

A Hardware

The experiments utilized a high-performance computing cluster with the above specifications. Training and evaluating 100 model instances across all datasets and architectures required substantial computational resources to ensure reliable estimation of means and standard deviations.

CPU Cores	128
CPU Memory	2 TB
GPU	8 × NVIDIA A100 80GB
Total GPU Hours	500

B Related work

Predictive uncertainty in LLMs can be estimated using various methods. Among these, *information-based* methods are widely utilized, which analyze token probability distributions by accessing logits or outputs from the internal layers of LLMs (Takayama and Arase, 2019; Fomicheva et al., 2020; van der Poel et al., 2022; Colombo et al., 2023), or by relying on the generated text (Tian et al., 2023). However, these methods are often outperformed by *sample diversity* methods, which involve generating multiple outputs for LLM and either aggregating their confidence scores or assessing their diversity (Kuhn et al., 2023; Malinin and Gales, 2021; Duan et al., 2024).

In contrast, *density-based* methods approximate the distribution of training data using embeddings of training instances (Lee et al., 2018; Yoo et al., 2022; Ren et al., 2023; Vazhentsev et al., 2023). Additionally, predictive uncertainty can be estimated using *reflexive* methods, in which the model is asked directly to provide confidence levels for its responses (Kadavath et al., 2022; Tian et al., 2023).

C Details on baseline UE methods

C.1 Pooling

Probing methods extend beyond representational analysis to sequence classification. A probe is an auxiliary classifier applied to intermediate hidden states to test whether internal representations encode information consistent with the conditioning prompt. We consider two probes: (i) a linear probe and (ii) an attention-pooling probe. For each model, we attach probes at three depths: *beginning* (≈25% of layers), *middle* (≈50%), and *last* (final layer).

Linear Probing. A linear probe classifies a single hidden state vector from a chosen sublayer (Kadavath et al., 2022). Let x_i be the i-th token and $\mathbf{h}_i \in \mathbb{R}^n$ its hidden representation. We assign $y_i \in \{0, 1\}$ to indicate whether x_i belongs to a wrong classified sample or not. The prediction is given by

$$p(y_i = 1 \mid \mathbf{x}_{\leq i}) = \sigma(\mathbf{w}^\top \mathbf{h}_i + b),$$

where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are learned parameters, and σ is the logistic function.

Attention-Pooling Probe. To include contextual information, we use an attention-pooling variant (Sky et al., 2024) that aggregates prior hidden states:

$$p(y_i = 1 \mid \mathbf{x}_{\leq i}) = \sigma(\mathbf{w}^\top \bar{\mathbf{h}}_i),$$

where

$$\bar{\mathbf{h}}_i = \sum_{j=1}^{i} \alpha_{i,j} \mathbf{h}_j, \quad \alpha_{i,j} = \frac{\exp(\mathbf{q}^\top \mathbf{h}_j)}{\sum_{k=1}^{i} \exp(\mathbf{q}^\top \mathbf{h}_k)},$$

and $\mathbf{q} \in \mathbb{R}^n$ is a learnable query vector.

C.2 Details on distance-based UE methods

Here, we describe the implementation as well as extensions of MD — a Mahalanobis distance-based uncertainty estimation method.

General description The MD-based uncertainty estimation method (Lee et al., 2018) models each class as a Gaussian distribution characterized by class centroids $\boldsymbol{\mu}_c = \mathbb{E}[h(\mathbf{x})|y = c]$, a shared covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{h} - \boldsymbol{\mu}_c)(\mathbf{h} - \boldsymbol{\mu}_c)^T]$.

The uncertainty score for an input $\mathbf{x}$ is given by the minimum Mahalanobis distance to any class centroid:

$$U_{\text{MD}}(\mathbf{x}) = \min_{c \in \mathcal{C}} (\mathbf{h} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1} (\mathbf{h} - \boldsymbol{\mu}_c). \quad (8)$$

where $\mathbf{h}$ is a representation from the penultimate layer.

Two methods extend MD to make it more robust: the **Relative Mahalanobis Distance (RMD)** (Ren et al., 2021) and the **Robust Distance Estimation (RDE)** (Yoo et al., 2022). These modifications enhance robustness to outliers while preserving the discriminative power of the original representations. **(MDM) Marginal Mahalanobis** distance measures how far a feature vector $\mathbf{h}$ is from the class-agnostic (marginal) background distribution of training features.

	LLaMA					Qwen					Avg.
	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	rank
Rand CLS. BCE loss	63.2±5.95	72.1±2.48	88.1±1.67	54.2±1.08	63.1±9.03	77.3±3.88	71.6±1.24	**86.7±1.28**	52.6±2.43	58.0±7.31	4.9
Rand linear. BCE loss	61.9±6.41	75.7±1.2	88.0±1.55	53.2±3.51	53.5±7.85	70.5±11.47	71.7±1.44	86.1±2.08	54.8±1.41	54.5±8.78	5.7
Trained CLS. BCE	**78.8±0.52**	88.5±0.39	88.6±1.41	60.8±1.94	77.9±2.23	80.2±2.0	**88.0±0.24**	86.5±1.63	62.4±1.92	74.5±2.03	2.7
Trained CLS. BCE+L2SP	78.5±0.6	88.5±0.44	88.7±1.52	61.1±1.88	**78.2±2.05**	80.2±1.75	88.0±0.3	86.3±1.73	**62.8±1.99**	**75.1±1.77**	2.9
Trained CLS. BCE+reg	78.6±0.81	**88.7±0.37**	88.8±1.57	62.3±1.37	77.9±2.25	80.0±1.99	88.0±0.29	86.6±1.73	62.3±1.85	74.6±2.07	2.6
AdUE	78.7±0.48	88.6±0.42	**88.9±1.62**	**62.8±1.22**	77.7±2.46	**80.5±1.4**	88.0±0.29	86.5±1.57	62.8±1.38	75.0±1.76	**2.2**

	Electra					Roberta					Avg.
	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	rank
Rand CLS. BCE loss	65.7±8.44	82.3±0.8	**88.6±0.89**	55.3±2.35	66.1±9.77	61.7±5.99	78.5±0.82	72.9±12.65	57.4±3.19	59.5±6.74	4.8
Rand linear. BCE loss	62.9±5.48	77.3±0.78	80.8±16.84	55.4±2.24	58.6±8.45	64.2±2.8	75.2±0.74	62.2±9.54	57.0±2.04	59.3±7.56	5.8
Trained CLS. BCE	80.2±0.99	85.6±0.41	87.0±2.6	61.5±1.58	74.8±0.67	75.4±2.97	85.9±0.77	**81.7±2.2**	61.7±0.73	74.5±0.42	2.8
Trained CLS. BCE+L2SP	80.1±0.99	84.7±0.52	87.3±2.46	62.2±0.85	74.4±1.04	75.3±2.92	86.0±0.14	81.5±1.95	61.6±1.4	**74.6±0.3**	3.0
Trained CLS. BCE+reg	80.1±1.07	**85.7±0.5**	87.2±2.51	62.2±1.26	74.9±0.84	**75.6±2.82**	85.9±0.78	81.5±2.19	**61.8±1.03**	74.4±0.49	2.4
AdUE	**80.4±0.89**	85.0±0.46	87.1±2.79	**62.5±1.13**	**74.9±0.39**	75.5±2.84	**86.0±0.22**	81.5±1.98	61.6±1.27	74.5±0.46	**2.2**

Table 2: Mean ROC—AUC and standard deviation over 5 runs are reported. Results indicate that using all loss components combined with classification head initialization yields the best performance.

Implementation details For Electra and Roberta models incorporate spectral normalization (Miyato et al., 2018) in the penultimate layer of their classification heads, which have been shown to stabilize density-based uncertainty estimation (Vazhentsev et al., 2022) and also test L2 normalization for representation vector $\mathbf{h}$, results shown in Tables 5, 6.

C.3 SRFT. Softmax response fine-tune

SRFT follows the same architecture and training pipeline as **AdUE**, but uses the Softmax Response Eq 1 as the base uncertainty instead of entropy. Directly optimizing the non-smooth max is unstable, so we replace it with a differentiable log-sum-exp approximation:

$$\mathrm{SmoothMax}_\lambda(\mathbf{p}) = \frac{1}{\lambda}\log\sum_{c=1}^{C}\exp\big(\lambda\, p_c\big), \lambda > 0, \tag{9}$$

which satisfies $\mathrm{SmoothMax}_\lambda(\mathbf{p}) \to \max_c p_c$ as $\lambda \to \infty$. We define the SRFT uncertainty as

$$U^{\mathrm{SRFT}}(\mathbf{x}) = 1 - \mathrm{SmoothMax}_\lambda\big(\mathbf{p}(\mathbf{x})\big), \tag{10}$$

.

Initialization and training. The SRFT head is initialized by copying the classifier's weights $\theta_{\mathrm{AdUE}} \leftarrow \theta_{\mathrm{init}}$ and is trained with the same three-term objective as AdUE (Eq. 7). In the output-consistency term, we replace $U_{\mathrm{Entropy}}(\mathbf{x})$ with the SR target $1 - \max_c p(c \mid \mathbf{x})$ (or its smoothed counterpart $1 - \mathrm{SmoothMax}_\lambda(\mathbf{p}(\mathbf{x}))$).

D Hyperparameters search

Adue. We conducted extensive hyperparameter optimization across the following ranges:

Probability entropy fine-tune parameters:

- Output regularization (α): 0.01, 0.1, 1.0
- L2SP (β): 0.01, 0.1, 1.0
- Learning rate: 4×10^{-4}, 1×10^{-4}, 1×10^{-5}
- Training Epochs: 20

Probe.

- **Linear probe**:
  - Learning rate: 1×10^{-3}, 4×10^{-4}, 1×10^{-4}
  - Training epochs: 20
- **Attention pooling probe**:
  - Learning rate: 1×10^{-3}, 4×10^{-4}, 1×10^{-4}
  - Training epochs: 20

E Datasets

We evaluated our approach on five benchmark datasets spanning different text classification tasks. The sizes of the datasets and the number of classes represented are shown in the Table 3.

SST-2 & SST-5 The Stanford Sentiment Treebank provides sentence-level sentiment labels from movie reviews. SST-2 is a binary classification task (positive/negative), while SST-5 offers fine-grained sentiment analysis (very negative to very positive). Both datasets contain parse trees enabling full-sentence compositional analysis.

20Newsgroups A classic text categorization corpus comprising newsgroup posts across 20 topics. The dataset presents challenges in document-level understanding and contains a significant class imbalance.

ToxiGen A large-scale dataset for detecting hate speech against 13 minority groups, with synthetic and human-annotated examples. It focuses on implicit toxicity detection in diverse linguistic constructions.

CoLA The Corpus of Linguistic Acceptability evaluates models' ability to judge grammatical correctness. It contains expert-labeled examples of English grammatical phenomena, testing linguistic competence.

F LoRA Adapter training

All models in our experiments employed parameter-efficient fine-tuning through Low-Rank Adaptation (LoRA). This approach enables effective model adaptation while maintaining computational efficiency and preserving the original model knowledge. The LoRA implementation follows standard practices for transformer-based architectures, with adaptations applied to attention mechanisms throughout the network. The specifications were as follows:

$$\text{LoRA config} = \begin{cases} \text{Weights} = \{W_q, W_k, W_v\} \\ \alpha = 16 \\ \text{rank} = 8 \\ \text{dropout} = 0.05 \end{cases}$$

Parameter	Value
Optimizer	AdamW
Learning rate	5×10^{-4}
Weight Decay	0.1
Batch Size	64
Learning Rate Schedule	Linear decay to 0
Warmup steps	0.1
Mixed-precision	bfloat16-mixed

F.1 Metrics

The final quality of each LoRA fine-tuned model is summarized in Table 4, which reports the mean and standard deviation of accuracy across multiple runs.

	News	SST2	SST5	Cola	Toxigen
LLaMA	77.5±2.2	**96.4±0.2**	**59.3±0.8**	**85.7±0.5**	**85.3±0.4**
Qwen	**79.3±0.7**	96.0±0.6	58.8±0.4	84.6±1.1	85.3±0.7
Electra	72.3±0.4	93.9±0.4	56.0±1.1	85.1±0.7	81.3±0.4
Roberta	74.2±0.7	93.2±0.5	55.2±1.2	83.3±0.9	79.4±1.0

Table 4: Classification accuracy across all datasets shows that the fine-tuned LLaMA model achieves the best performance among all the models we trained.

G Ablation

We also perform an ablation study to examine: (a) the effect of the head architecture and initialization, and (b) the contribution of each loss term. We compare our approach of using the full classification layer (with $|C|$ outputs) initialized from the original model vs. two alternatives: using a randomly-initialized classifier layer, and using a simpler single-output linear layer on $\mathbf{h}$ (logistic regression) initialized randomly. All are trained with the BCE loss. We further ablate the loss by removing the L2-SP term, the entropy regularization term, or both, retraining AdUE in each setting.

- **Trained CLS. BCE** — Initialize the uncertainty head with θ_{init} (copied from the trained classification head), apply the entropy mapping, and train using only binary cross-entropy (BCE) on the per-sample error label.
- **Trained CLS. BCE+L2SP** — Same initialization and entropy mapping as above; train with BCE plus L2-SP weight anchoring to θ_{init} (i.e., $\|\theta - \theta_{\text{init}}\|_2^2$).
- **Trained CLS. BCE+Reg** — Same initialization and entropy mapping; train with BCE plus an output-consistency regularizer that keeps $U_{\text{AdUE}}(x)$ close to the original entropy $U_{\text{Entropy}}(x)$ (i.e., $(U_{\text{AdUE}}(x) - U_{\text{Entropy}}(x))^2$).
- **Rand Linear. BCE** — Replace the head with a randomly initialized single-output linear layer on z; train with BCE to predict the per-sample error indicator.
- **Rand CLS. BCE** — Randomly initialize a head with the same architecture as the classification head (shape of θ_{init}), include the entropy mapping, and train with BCE to predict the error indicator.

Results are shown in Table 2. We find that using the original classification head initialization provides a significant benefit – using a random head

Dataset	Description	Classes	Train size	Valid size	Test size
SST-2	Stanford Sentiment Treebank (Socher et al., 2013)	2	53879	13470	872
SST-5	Stanford Sentiment Treebank (Socher et al., 2013)	5	6840	1711	1043
20Newsgroups	Topic classification (Lang, 1995)	20	9051	2263	7532
ToxiGen	Hate speech detection (Hartvigsen et al., 2022)	2	7168	1792	940
CoLA	Linguistic acceptability (Warstadt et al., 2019)	2	6840	1711	1043

Table 3: Dataset information, number of classes, and split sizes.

Model			Electra					Roberta			
Dataset	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	Rank ↓
RMD	70.5±2.4	**83.7±0.6**	80.1±2.1	**57.3±1.2**	68.6±0.9	64.2±1.7	84.3±0.7	73.9±3.8	57.9±1.1	68.3±2.5	2.8
RMD L2 norm	70.0±2.6	**83.7±0.6**	79.2±4.1	57.2±1.3	67.9±1.5	65.3±1.7	84.7±0.7	73.0±2.8	**58.1±1.1**	**69.1±2.2**	2.9
MD	76.2±2.0	74.6±0.7	85.1±2.4	53.3±1.1	62.8±3.9	52.8±11.2	78.4±0.7	76.4±11.7	51.0±1.8	53.2±11.3	4.0
MD L2 norm	**80.9±1.1**	74.1±0.7	**88.2±1.0**	55.5±1.0	74.0±1.6	**73.7±0.6**	79.5±0.4	**82.8±2.2**	55.6±1.9	65.1±9.2	2.3
RDE	73.2±2.6	68.5±0.5	85.3±2.2	53.8±1.3	64.8±4.0	50.6±11.3	74.2±1.0	74.8±11.5	51.0±1.7	51.6±11.0	4.6
RDE L2 norm	80.0±1.3	70.8±1.8	68.8±24.1	**56.2±0.9**	**74.3±1.3**	73.4±1.4	77.3±0.5	67.3±25.4	55.9±1.5	65.1±8.9	3.5

Table 5: L2 normalization on representation show better UE ROC—AUC for encoder models

Model			LLaMA					Qwen			
Dataset	Cola	News	SST2	SST5	Toxigen	Cola	News	SST2	SST5	Toxigen	Rank ↓
RMD	67.0±5.6	80.4±1.1	**79.0±2.3**	**53.4±0.4**	**67.6±2.5**	**65.3±4.2**	**82.0±0.8**	78.5±4.6	54.3±1.5	65.8±1.5	1.4
RMD L2 norm	**67.3±6.0**	**80.7±1.1**	78.7±2.5	**53.4±0.4**	67.5±2.5	65.2±4.1	81.2±0.7	**78.6±4.1**	54.3±1.4	**66.4±1.9**	1.4
MD	60.5±4.0	55.2±2.4	57.3±12.2	52.0±0.9	55.7±2.2	52.1±3.8	60.5±1.6	52.6±6.7	52.2±1.5	52.7±3.0	4.0
MD L2 norm	60.5±4.3	56.3±2.7	59.9±10.0	51.4±1.0	56.3±1.8	56.7±2.6	53.9±0.7	54.2±6.4	52.1±1.2	58.1±5.0	3.7
RDE	63.7±9.6	43.5±3.7	60.0±9.3	51.6±1.2	52.7±4.7	42.5±12.0	51.0±4.4	52.2±7.4	52.1±1.0	51.3±2.8	4.8
RDE L2 norm	62.1±5.8	47.6±3.3	72.2±8.7	49.2±1.8	53.6±2.0	47.2±2.4	48.2±2.8	63.8±16.2	48.0±0.9	54.9±5.5	4.4

Table 6: For generative model L2 normalization on representation do not change significant ROC—AUC

(Rand CLS head) or a single-output probe (Rand linear) yields worse performance (higher average rank). This suggests that starting from a strong, pre-trained head is advantageous. In terms of loss terms, we observe that removing any one of the regularization terms degrades performance: the full AdUE (with BCE+reg+L2SP) achieves the best mean rank (2.2), whereas leaving out regularization (BCE only) or using only one regularizer leads to higher (worse) ranks. Notably, adding the entropy regularizer (BCE+reg) helps more than adding L2-SP alone (BCE+L2SP), as seen by the ranks 2.5 vs 3.0.