Generative or Discriminative? Revisiting Text Classification in the Era of Transformers

Siva Rajesh Kasa [1] **Karan Gupta** [1] **Sumegh Roychowdhury** [1] **Ashutosh Kumar** [1] **Yaswanth Biruduraju** [1]
Santhosh Kumar Kasa [1] **Nikhil Priyatam Pattisapu** [1] **Arindam Bhattacharya** [1] **Shailendra Agarwal** [2]
Vijay huddar [1]

Abstract

In text classification, the classical comparison between discriminative and generative classifiers gains renewed relevance in the transformer era, where computational constraints often limit thorough experimentation. Through systematic small-scale experiments on text classification tasks, we investigate how the fundamental "two regimes" phenomenon—where generative classifiers excel with limited data but show higher asymptotic error—manifests across modern architectures (Auto-regressive, Masked Language Models, Discrete Diffusion, and Encoders). By training models from scratch on controlled text datasets, we isolate and analyze core architectural behaviors in terms of sample efficiency, calibration, and preservation of ordinal relationships. Our findings provide insights into the inherent trade-offs of different modeling approaches for text classification, demonstrating how small-scale experimentation can inform both theoretical understanding and practical architectural choices.

1. Introduction

Text Classification (TC), a fundamental task in Natural Language Processing (NLP), provides an ideal testbed for systematic investigation of machine learning principles. Since the emergence of transformer architectures, the field has been dominated by discriminative classifiers that leverage token embeddings (e.g., the `[CLS]` token in BERT (Devlin et al., 2019)). These models directly learn the conditional probability distribution $P_\theta(y|X)$, where X denotes the input text and y represents the ground truth label. However, as these discriminative models grow larger, systematic experimentation becomes prohibitively expensive, making it challenging to understand their behavior in real-world scenarios where labeled data is scarce or expensive to obtain (Zheng et al., 2023). On the other hand, generative classifiers, which model the joint distribution $P_\theta(X, y)$, are known to work better in low-data settings, giving rise to the classical 'two-regimes' phenomenon (Ng & Jordan, 2001; Yogatama et al., 2017; Zheng et al., 2023; Li et al., 2025). This advantage stems from their ability to learn underlying data distributions rather than just decision boundaries. The inherent data efficiency of generative approaches, combined with recent advances such as Discrete Diffusion (Lou et al., 2024), motivates us to revisit the classical discriminative versus generative debate through controlled, small-scale experiments.

Our **main contributions** include: **(a)** A systematic small-scale comparative study of discriminative (Encoder) and generative (Text Diffusion, AR, MLM) approaches across 9 classification benchmarks, revealing nuanced interactions between model size and sample complexity. **(b)** Comprehensive analyses of model scaling behavior, sample efficiency, and performance in low-resource settings, including novel evaluation perspectives examining ordinal relationships and calibration. **(c)** Practical recommendations for model selection based on deployment constraints and requirements.

2. Related Work

The study of generative versus discriminative classifiers originated with Efron (1975)'s analysis of logistic regression and discriminant analysis, with Ng & Jordan (2001) establishing the fundamental trade-off between generative models' faster

[1]Amazon, Inc. [2]Google, India. Correspondence to: Siva Rajesh Kasa <kasasiva@amazon.com>, Karan Gupta <karaniis@amazon.com>, Sumegh Roychowdhury <sumegr@amazon.com>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

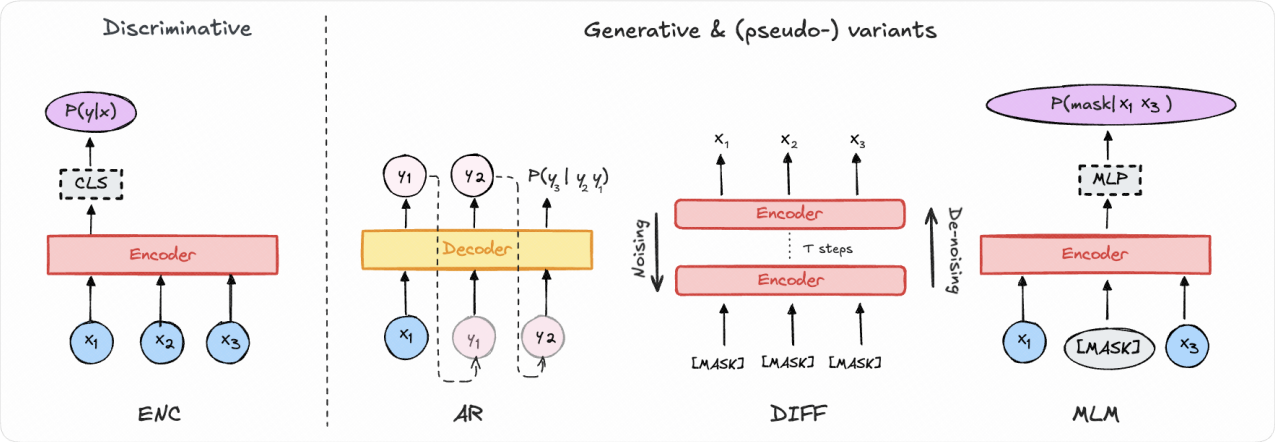

Figure 1: **[Best viewed in color]** Illustration of different modeling paradigms (`ENC`: Encoder-based classification, `MLM`: Masked Language Modeling, `AR`: Auto-Regressive Model, `DIFF`: Discrete Text Diffusion).

learning rate and discriminative models' lower asymptotic error. Yogatama et al. (2017) provided the first neural investigation using LSTMs for text classification, demonstrating a true generative approach by prepending the label y to the input text X and maximizing $P(X|y)P(y)$ through the class conditional likelihood $P(X|y) = \prod_{t=1}^{T} p(x_t|\boldsymbol{x}< t, y)$. This traditional generative formulation requires #label forward passes during inference as it computes $P(X|y)$ for each possible label. While Li et al. (2025) showed that appending labels at the end of sequences can yield better in-distribution performance and enable single-pass inference, this represents a pseudo-generative variant as it doesn't strictly model $P(X|y)$. Similar pseudo-generative approaches include MLM (Devlin et al., 2019) and Discrete Diffusion (Lou et al., 2024), with recent work (Sahoo et al., 2024) revealing connections between these objectives. These approaches can be viewed through the lens of multi-task learning, where joint modeling of $P(X, y)$ involves balancing unsupervised and supervised objectives (Wu et al., 2020; Hu et al., 2023). Zheng et al. (2023) extended the theoretical understanding to multi-class and non-linear settings, while recent studies have highlighted generative classifiers' robustness to distribution shifts (Li et al., 2025; Stanley et al., 2025). Refer to Appendix A for a more detailed review on Discriminative vs Generative and Pseudo-generative variants, their connection to multi-task learning, Discrete Diffusion, Ordinality & Calibration.

3. Methodology

We approach text classification through two paradigms: **(a) Generative** - Discrete Diffusion models, Auto-regressive models (AR), and Masked Language Models (MLM) & **(b) Discriminative** - Encoder-based transformer models. Let $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$ denote the dataset where X_i is the input text and $y_i \in \mathcal{Y}$ is the corresponding label. We experiment with four different architectures.

(1) Encoder-based classification (`ENC`): A standard transformer encoder that processes input text to produce embeddings, followed by a linear classification head trained with cross-entropy loss.

(2) Masked Language Modeling (`MLM`): Following BERT (Devlin et al., 2019), we mask 15% of input tokens during training and predict them using bidirectional context. While not explicitly modeling $P(X|y)$, MLM approximates pseudo-likelihood through $P(x_m|x_{\setminus m})$ (Wang & Cho, 2019). At training, we use the template "`[CLS]` Input `[SEP]` The label is `[Label]`" and at inference, we replace `[Label]` with a `[Mask]` and predict it by restricting it to $\mathcal{Y}$.

(3) Auto-regressive modeling (`AR`): Following GPT (Radford et al., 2018), we train a causal model to predict next tokens given previous ones. For true generative variants (`AR`), labels are prepended and the model explicitly learns $P(X|y)P(y)$, requiring $\#_{\text{label}}$ forward passes at inference (Yogatama et al., 2017; Li et al., 2025). In pseudo-generative variants ($\texttt{AR}_{pseudo}$), labels are appended, enabling single-pass inference but no longer strictly modeling the joint distribution(Li et al., 2025). This label placement distinction is only relevant for causal architectures like AR with left-to-right attention; it has no theoretical impact on bidirectional models like MLM or DIFF that can attend to the full context.

(4) Text Diffusion (`DIFF`): Following Lou et al. (2024), we gradually corrupt text to `[MASK]` tokens and train a model to

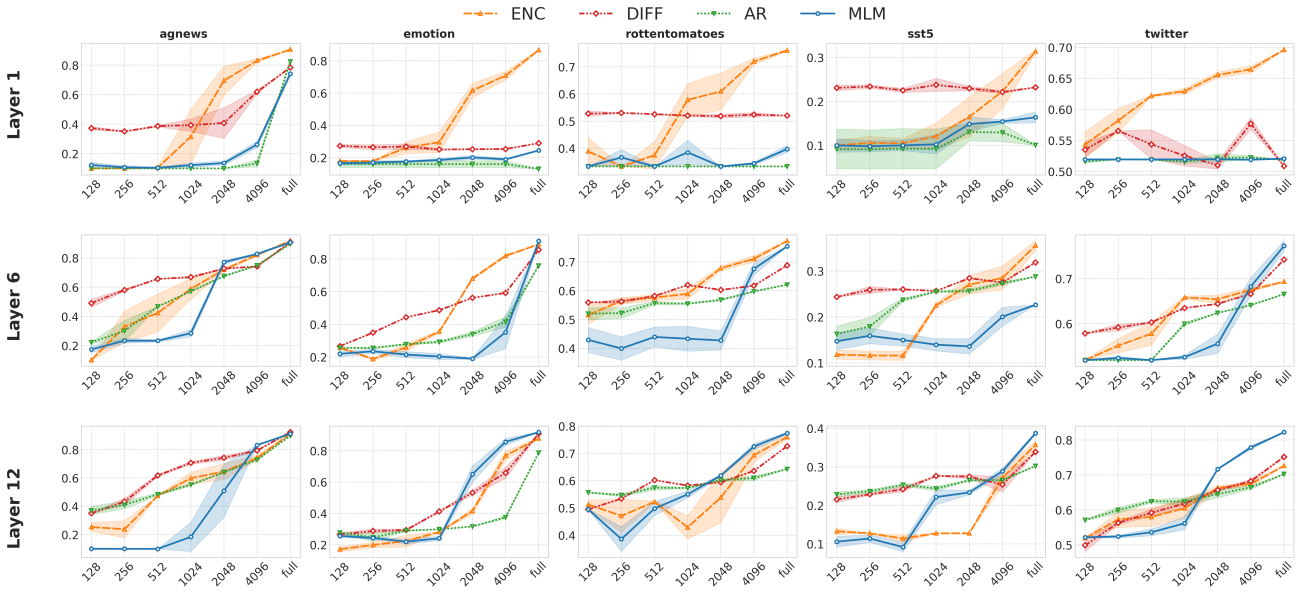

Figure 2: **[Best viewed in color]** Comparison of weighted-F1 scores of models across different configurations (↑ is better). For rest of the datasets, refer to Figure 7 in Appendix F. (`X-axis`: sample size, `Y-axis`: weighted-F1 score)

reverse this corruption process. At inference, we use the template "Input `[SEP]` The label is `[MASK]`" and predict the masked label token. Like MLM, this approach is inherently pseudo-generative as it models token reconstruction rather than true $P(X|y)$. Refer to Appendix B for a detailed overview of each approach.

4. Experiments and Results

Our study investigates three key aspects: (1) comparative performance of different modeling approaches trained from scratch across varying data and model sizes, (2) robustness to input perturbations through random token substitution and dropping, and (3) model calibration and preservation of ordinal relationships in predictions. We evaluate five modeling approaches (`AR`, $\texttt{AR}_{pseudo}$, `MLM`, `DIFF`, and `ENC`) across 9 text classification benchmarks spanning sentiment analysis, news categorization, and social media analysis: **AG News** (Zhang et al., 2015), **Emotion** (Saravia et al., 2018), **SST2 & SST5** (Socher et al., 2013), **Multiclass Sentiment**, **Twitter Financial News**, **IMDb** (Maas et al., 2011), and **Hate Speech** (Davidson et al., 2017). To ensure fair comparison, all approaches use the same transformer backbone architecture, varying only in their objective functions and label placement. For systematic evaluation, we experiment with three model scales: small (1 layer, 1 attention head), medium (6 layers, 6 heads), and large (12 layers, 12 heads). We vary dataset sizes from 128 to full data, apply input perturbations via random token dropping and substitution, and measure performance using weighted-F1 scores. For ordinal relationship assessment, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE), while for calibration quality we measure Expected Calibration Error (ECE), Maximum Calibration Error (MCE), and Unimodality (UM). All experiments are repeated with 3 random seeds, totaling 2835 configurations. Implementation details are provided in Appendix C.

4.1. Results and Key Findings

For **1-layer, 1-head** models (Figure 2), all approaches show near-random performance in low-data regimes. However, as training data increases, only `ENC` (orange line) continues to improve, ultimately outperforming others in high-data settings. This suggests that **for small models - often necessary due to real-world latency constraints - `ENC` is the most effective approach.** The classical 'two regimes' phenomenon does not manifest when the model size is small. The pattern shifts dramatically for larger architectures.Under the **12-layer, 12-head** configuration, both generative models—`AR` and `DIFF`—outperform `ENC` in low-data settings, with this advantage diminishing as data increases. This aligns with previous findings (Ng & Jordan, 2001; Yogatama et al., 2017; Rezaee et al., 2021) about generative models' advantages in data-limited scenarios. Surprisingly, for large models, the pseudo-generative `MLM` (*blue* line) consistently outperforms all methods across

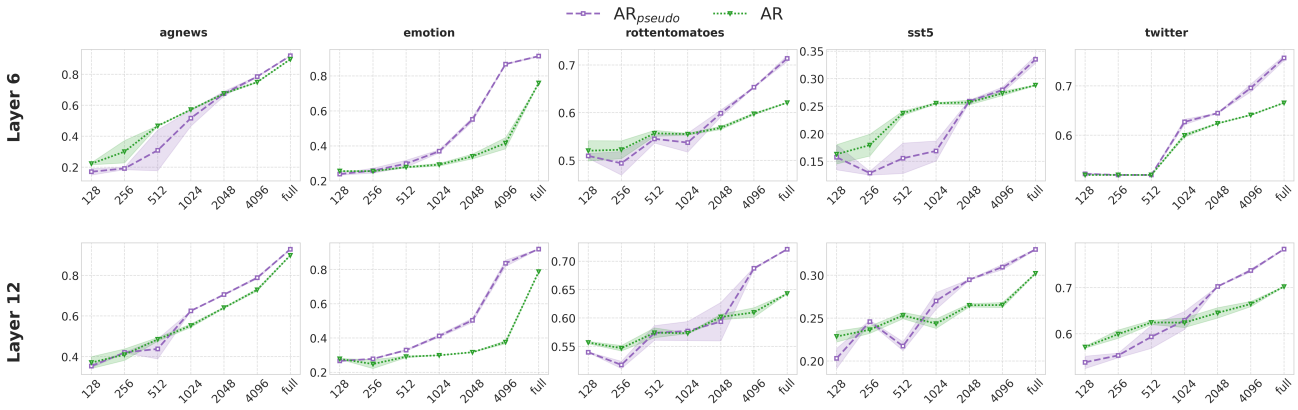

Figure 3: **[Best viewed in color]** Comparison of weighted-F1 scores between $\texttt{AR}_{pseudo}$ and AR (↑ is better). 1-layer results are omitted here as they are mostly trivial in low-data settings. Results for remaining datasets are provided in Figure 8, Appendix F. (`X-axis`: sample size, `Y-axis`: weighted-F1 score)

our 9 benchmark datasets in high-data settings, **challenging the conventional wisdom about discriminative dominance in high-sample regime.** This aligns with Erhan et al. (2010)'s finding that pseudo-generative models implicitly perform unsupervised pre-training alongside supervised learning, creating an effective multi-task setup (Section 2). Thus, **for scenarios without model size constraints, generative models emerge as the optimal choice for low-data settings** such as for low-resource languages and continual learning applications requiring frequent updates with limited samples, while **pseudo-generative MLM is superior when abundant labeled data is available.** Coming to the medium scale **6-layer, 6-head** configuration, in low-data settings, DIFF emerges as the best performing model across all datasets, clearly outperform even it's generative counterpart AR. As the training data size increases, we see that the discriminative ENC outperforming DIFF. Thus, **in medium scale architectures, between the generative DIFF and the discriminative ENC, the classical 'two regimes' still holds.** Figure 3 shows that $\texttt{AR}_{pseudo}$ generally underperforms AR and also displays **higher variance** in low-data settings—the recommended use case—while the opposite holds in high-data scenarios. This reveals a new insight beyond Li et al. (2025), who only evaluated full-data settings where $\texttt{AR}_{pseudo}$ performed better in-distribution. As noted in Section B, AR requires $|label|$-times forward passes per prediction, unlike the single pass needed for $\texttt{AR}_{pseudo}$; however, this can be mitigated via batching or parallel processing, reducing inference time differences at the cost of higher computation.

Figures 4,5,6 in Appendix E presents ordinal and calibration results. DIFF does not support calibration metrics like ECE, MCE, and UM, as its masking/absorbing noise process produces only binary outputs rather than soft probabilities. While a uniform noise schedule can yield probabilities over $\mathcal{V}$, it performed slightly worse, so we used the absorbing schedule in our study. From the ECE and MCE plots, we observe that ENC outputs remain well-calibrated across all sample sizes, while MLM reaches similar calibration only in high-data regimes. We also see that MLM and ENC achieve UM in over 80% of the samples, aligning with findings from Kasa et al. (2024). Their MAE and MSE values are also low, indicating strong ordinality in high-data settings. This completes the picture for *large* models under high-data, where MLM not only outperforms others in weighted-F1 but is also well-calibrated and ordinal, making it a strong candidate for real-world deployment. However, under low-data conditions, 12-layers AR outperforms $\texttt{AR}_{pseudo}$ in 7 out of 9 datasets on calibration metrics. It also surpasses DIFF in ordinal performance, thus making it the more reliable choice among generative models in low-data scenario. Also, even though generative approaches like DIFF were recommended based on weighted-F1 for 6-layers case (in Figure 2) deploying them in production could be risky when calibrated or ordinal probabilities are required, especially for imbalanced datasets like *twitter* and *hatespeech* (see Appendix E). These metrics are particularly important when downstream models consume output probability scores as features which is often the case in multi-stage ranking systems.

5. Conclusion

Our study offers practical modeling recommendations across deployment scenarios. For latency-sensitive applications, ENC is ideal—especially in the 1-layer setting—due to its accurate, well-calibrated, ordinal outputs. For offline settings with sufficient data, the 12-layer MLM performs best across F1, calibration, and ordinal metrics. In low-resource scenarios, both AR and DIFF are strong options, with DIFF favored for its performance at 6-layers. However, if calibrated probability outputs

are essential, such as in ranking pipelines, AR is the preferred choice.

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Blasiok, J., Gopalan, P., Hu, L., and Nakkiran, P. When does optimizing a proper loss yield calibration? *Advances in Neural Information Processing Systems*, 36:72071–72095, 2023.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pp. 512–515, 2017.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Efron, B. The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70:892–898, 1975. URL https://api.semanticscholar.org/CorpusID:34806014.

Erhan, D., Courville, A., Bengio, Y., and Vincent, P. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 201–208. JMLR Workshop and Conference Proceedings, 2010.

Hayashi, H. A hybrid of generative and discriminative models based on the gaussian-coupled softmax layer. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2894–2904, 2025. doi: 10.1109/TNNLS.2024.3358113.

He, Z., Sun, T., Tang, Q., Wang, K., Huang, X., and Qiu, X. DiffusionBERT: Improving generative masked language models with diffusion models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4521–4534, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.248. URL https://aclanthology.org/2023.acl-long.248/.

Hu, Y., Xian, R., Wu, Q., Fan, Q., Yin, L., and Zhao, H. Revisiting scalarization in multi-task learning: A theoretical perspective. *Advances in Neural Information Processing Systems*, 36:48510–48533, 2023.

Jaini, P., Clark, K., and Geirhos, R. Intriguing properties of generative classifiers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rmg0qMKYRQ.

Kasa, S. R., Goel, A., Gupta, K., Roychowdhury, S., Priyatam, P., Bhanushali, A., and Srinivasa Murthy, P. Exploring ordinality in text classification: A comparative study of explicit and implicit techniques. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5390–5404, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.320. URL https://aclanthology.org/2024.findings-acl.320/.

Kumar, A., Ahuja, K., Vadapalli, R., and Talukdar, P. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345, 2020. doi: 10.1162/tacl_a_00318. URL https://aclanthology.org/2020.tacl-1.22/.

Li, A. C., Kumar, A., and Pathak, D. Generative classifiers avoid shortcut solutions. In *The Thirteenth International Conference on Learning Representations*, 2025.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022a.

Li, X. et al. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Li, Y., Bradshaw, J., and Sharma, Y. Are generative classifiers more robust to adversarial attacks? In *International Conference on Machine Learning*, pp. 3804–3814. PMLR, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In Lin, D., Matsumoto, Y., and Mihalcea, R. (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

McCallum, A., Pal, C., Druck, G., and Wang, X. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *AAAI*, volume 1, pp. 6, 2006.

Merkle, E. C. and Steyvers, M. Choosing a strictly proper scoring rule. *Decision Analysis*, 10(4):292–304, 2013.

Ney, H., Essen, U., and Kneser, R. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38, 1994.

Ng, A. Y. and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, 2001.

Pang, B. and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Knight, K., Ng, H. T., and Oflazer, K. (eds.), *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL https://aclanthology.org/P05-1015/.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *OpenAI*, 2018.

Raina, R., Shen, Y., Mccallum, A., and Ng, A. Classification with hybrid generative/discriminative models. *Advances in neural information processing systems*, 16, 2003.

Rezaee, M., Darvish, K., Kebe, G. Y., and Ferraro, F. Discriminative and generative transformer-based models for situation entity classification. *arXiv preprint arXiv:2109.07434*, 2021.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL https://www.aclweb.org/anthology/D18-1404.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Stanley, E. A., Forkert, N. D., and Wilms, M. Does a diffusion-based generative classifier avoid shortcut learning in medical image analysis? an initial investigation using synthetic neuroimaging data. In *Medical Imaging 2025: Imaging Informatics*, volume 13411, pp. 94–99. SPIE, 2025.

Sun, H., Yu, L., Dai, B., Schuurmans, D., and Dai, H. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Teh, Y. W. A bayesian interpretation of interpolated kneser-ney nus school of computing technical report tra2/06. *National University of Singapore*, pp. 1–21, 2006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Wang, A. and Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In Bosselut, A., Celikyilmaz, A., Ghazvininejad, M., Iyer, S., Khandelwal, U., Rashkin, H., and Wolf, T. (eds.), *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304. URL https://aclanthology.org/W19-2304/.

Wu, S., Zhang, H. R., and Ré, C. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020.

Xue, J.-H. and Titterington, D. M. Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". *Neural Process. Lett.*, 28(3):169–187, December 2008. ISSN 1370-4621. doi: 10.1007/s11063-008-9088-7. URL https://doi.org/10.1007/s11063-008-9088-7.

Yogatama, D., Dyer, C., Ling, W., and Blunsom, P. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint*, 2017.

Zeng, J., Xu, J., Zheng, X., and Huang, X. Certified robustness to text adversarial attacks by randomized [mask]. *Computational Linguistics*, 49(2):395–427, 2023.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 649–657, Cambridge, MA, USA, 2015. MIT Press.

Zhang, X., Hong, H., Hong, Y., Huang, P., Wang, B., Ba, Z., and Ren, K. Text-crs: A generalized certified robustness framework against textual adversarial attacks. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2920–2938. IEEE, 2024.

Zheng, C., Wu, G., Bao, F., Cao, Y., Li, C., and Zhu, J. Revisiting discriminative vs. generative classifiers: Theory and implications. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42420–42477. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zheng23f.html.

A. More Background and Related works

Generative and Discriminative Models for Classification. The comparison between generative and discriminative classifiers originated with Efron (1975)'s analysis of logistic regression and discriminant analysis. Building on this foundation, Ng & Jordan (2001) examined naive Bayes and logistic regression, establishing the fundamental trade-off between generative models' faster learning rate and discriminative models' lower asymptotic error. Their theoretical analysis heavily depends on linearity and independence assumptions. However, subsequent work by Xue & Titterington (2008) challenged these findings through empirical studies and asymptotic analysis of statistical efficiency. Yogatama et al. (2017) provided the first empirical study of discriminative vs generative models for TC with neural architectures using LSTMs. They maximize the joint probability $P(X, y) = P(X|y)P(y)$ by concatenating the label y text at the beginning of the input text X and maximizing the class conditional likelihood i.e. $P(X \mid y) = \prod_{t=1}^{T} p\left(x_t \mid \boldsymbol{x}_{<t}, y\right)$. The final predicted label is obtained by $\hat{y} = \texttt{argmax}_y\, P(X|y)P(y)$. They found that generative LSTMs have better accuracy than their discriminative counterparts at low-sample regimes. Further, they noted that the neural generative LSTMs are generally better than baseline generative models with stronger independence assumptions (e.g. naive Bayes, Kneser–Ney Bayes (Ney et al., 1994; Teh, 2006)). Next, the work by Zheng et al. (2023) has extended the theoretical understanding of generative classifiers to multi-class and non-linear settings. More recent studies (Li et al., 2025; Stanley et al., 2025) have found that generative classifiers tend to avoid shortcut learning and exhibit greater robustness to distribution shifts.

While prior studies provide valuable insights, the landscape of NLP has evolved dramatically with the advent of novel transformer-based generative paradigms such as Auto-Regressive (AR) models (Radford et al., 2018) and Discrete Diffusion models (Lou et al., 2024). Our work extends beyond these previous comparisons by conducting the first comprehensive evaluation of modern transformer-based generative and discriminative classifiers for TC. While previous works primarily focused on classification accuracy and sample complexity, we examine multiple dimensions that are crucial for real-world deployments. For instance, Yogatama et al. (2017) initial work with neural architectures was limited to a fixed model size, leaving open questions about how the generative-discriminative trade-off varies with model capacity and computational budget—questions that have become increasingly relevant in the era of large language models. Similarly, though Zheng et al. (2023) provided theoretical insights for multi-class settings, their analysis did not address practical considerations like calibration quality or preservation of ordinal relationships between classes.

Pseudo-Generative Models. Recent work (Sahoo et al., 2024) highlights a natural connection between Discrete Text Diffusion (Lou et al., 2024) and the Masked Language Modeling (MLM) objective in BERT (Devlin et al., 2019), showing that the diffusion objective can be expressed as a weighted sum of MLM losses. Using transformer encoder models, this approach achieves likelihood bounds comparable to or better than those in Lou et al. (2024). Motivated by this, we include vanilla MLM as a baseline for text classification. While MLM has typically served as a pretraining objective followed by fine-tuning (Liu et al., 2019), there has been little systematic study of its direct use for classification. Although MLM does not explicitly model $P(X|y)$, it estimates $P(x_m|x_{\setminus m})$, where x_m is a masked token and $x_{\setminus m}$ represents all other tokens. This approximates the pseudo-likelihood of $P(X, y)$ when modeled over the corpus (Wang & Cho, 2019). We therefore classify MLM as a pseudo-generative model.

Also, traditional generative classifiers aim to model $P(X|y)$ by prepending the label token. However, recent work (Li et al., 2025) shows that appending the label at the end—though not strictly modeling $P(X|y)$—can yield better in-distribution performance. This setup also enables efficient inference, requiring only a single forward pass to predict the label, unlike traditional generative models that need $\#_{\text{label}}$ forward passes. These benefits motivate the inclusion of such pseudo-generative models in our benchmarks. Notably, these approaches involve minimal changes to standard transformer architectures—typically just altering label placement or the loss function—while preserving the core model design. This allows for fair comparisons using widely available implementations accessible to practitioners.

We also acknowledge a separate class of hybrid generative-discriminative models, where some subset of parameters are trained generatively and others discriminatively (Raina et al., 2003; McCallum et al., 2006; Hayashi, 2025). However, we exclude them from our study, as their architectural differences hinder fair comparison with fully generative or discriminative models, placing them outside the scope of this work.

Relation to Multi-task Learning. Learning $\log P(X, y)$ jointly, when factored as $\log P(X) + \log P(y|X)$ (or $\log P(y) + \log P(X|y)$) can be viewed as a multi-task learning setup, where unsupervised learning of $\log P(X)$ ($\log P(Y)$) and supervised learning of $\log P(Y|X)$ ($\log P(X|Y)$) represent two different but related tasks. This connection is supported by empirical results showing that unsupervised pre-training helps downstream supervised tasks (Erhan et al., 2010). As demonstrated by Wu et al. (2020); Hu et al. (2023), when model capacity is sufficiently large, such multi-task learning

setups tend to be more successful - the model has enough capacity to perform well on both the unsupervised and supervised objectives. However, with limited model capacity, there are inherent trade-offs between the tasks, leading to challenges in jointly optimizing for both $P(X)$ and $P(y|X)$ (or $P(y)$ and $P(X|y)$). This insight motivates us to conduct a systematic study examining the relationship between model capacity and the performance of discriminative vs generative classifiers - an analysis that has not been previously undertaken in the literature.

Discrete Diffusion Models for Classification. Recent advances in discrete diffusion models have shown promising results in text generation tasks, matching or surpassing autoregressive models at GPT-2 scale (Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). While these models have demonstrated success in controlled generation tasks (Li et al., 2022a; He et al., 2023), specifically syntax controlled generation of text (Kumar et al., 2020) and text infilling, their application to classification remains relatively unexplored. Traditional diffusion models for text generation, such as DiffusionBERT (He et al., 2023), DiffusionLM (Li et al., 2022b), and D3PM (Austin et al., 2021), operate by embedding discrete token sequences into continuous spaces and applying Gaussian noise-based diffusion. In contrast, SEDD (Lou et al., 2024) was the first to directly model diffusion in discrete space through a score entropy-driven objective. Hence, we adopt SEDD as our baseline method. Our work provides the first systematic evaluation of discrete diffusion models for classification tasks, comparing them against traditional discriminative and generative approaches.

Robustness to Noise. Previous studies have examined robustness primarily through the lens of adversarial attacks (Li et al., 2019), distribution shifts (Li et al., 2025) and domain shifts (Jaini et al., 2024). While recent work has provided certified robustness guarantees for perturbations like insertion, deletion, reordering and synonyms for specific architectures (Zeng et al., 2023; Zhang et al., 2024), our study presents comparisons across model families under two different noise conditions in the context of TC for transformer architectures.

Calibration & Ordinality. Model calibration is crucial in classification, as it reflects how well predicted probabilities align with actual frequencies. Proper Scoring Rules (PSR) (Merkle & Steyvers, 2013) offer a theoretical basis for producing calibrated predictions: a scoring rule (i.e. loss function) is proper if its expected value is minimized only when predicted probabilities match the true distribution. All our modeling approaches—Generative (AR, MLM, Discrete Diffusion) and Discriminative (Encoder)—optimize proper scoring rules. GPT and MLM maximize likelihood, Discrete Diffusion optimizes a variational bound, and cross-entropy minimizes the KL-divergence between predicted and true distributions. Recent work (Blasiok et al., 2023) shows that models trained with PSRs are often naturally calibrated when achieving low training loss, without requiring post-hoc calibration. This motivates us to empirically assess calibration across our models, as their differing architectures and objectives may still lead to varying calibration behaviors.

Ordinality in text classification is essential for applications like sentiment analysis or medical assessments, where label order affects decisions and distant misclassifications are more harmful. Recent works (Kasa et al., 2024) systematically compare *explicit* methods—like custom losses enforcing label order—with *implicit* approaches using pretrained models' semantics. However, no prior work focuses on exploring ordinality across diverse modeling frameworks trained from scratch.

B. Methodology

We approach the problem of label classification by leveraging two popular language modeling paradigms: **(a) Generative** - Discrete Diffusion models, Auto-regressive models (AR), and Masked Language Models (MLM) & **(b) Discriminative** - Encoder-based transformer models. Note that, for brevity, we use the term *"generative"* from this point onward to also include the **pseudo-generative** baselines. Let $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$ denote the dataset where X_i is the input text and $y_i \in \mathcal{Y}$ is the corresponding label from a finite set of classes $\mathcal{Y}$. Generative models tend to learn the joint data distribution $P(X, y)$ first and then try to infer the label using the marginals, whereas Discriminative models directly learn the conditional distribution $P(y|X)$. Note that each $X_i = x_i^1 \dots x_i^n$, where x_i^j is a token from the associated vocabulary $\mathcal{V}$.

B.1. Discriminative Model for Classification

(1) Encoder-based classification (`ENC`): A Transformer encoder (Vaswani et al., 2017) f_θ encodes the input as $h_i = f_\theta(X_i)$ as a d-dimensional embedding, followed by a linear classifier head $W \in \mathbb{R}^{|\mathcal{Y}| \times d}$ which is the standard discriminative learning setup:

$$\hat{y}_i = \texttt{softmax}(Wh_i), \quad \mathcal{L}_{\text{enc}} = -\sum_{i=1}^{N} \log P(y_i|X_i) \tag{1}$$

where $\mathcal{L}_{\text{enc}}$ is the cross-entropy based objective for training the encoder model.

B.2. Generative Models for Classification

(2) Masked Language Modeling (`MLM`): During training, we mask 15% of tokens in input sequence $X_i = x_i^1 \dots x_i^n$ following Devlin et al. (2019) and predict them using unmasked bi-directional context. Wang & Cho (2019) show that the MLM objective stochastically captures the *pseudo-loglikelihood* which makes it similar to a denoising autoencoder (Vincent et al., 2010). Hence, we consider MLM under the generative family of models. Formally, the objective is:

$$\mathcal{L}_{\text{mlm}} = -\sum_{i=1}^{N} \sum_{j \in \mathcal{M}_i} \log P(x_i^j | X_i^{\setminus j}) \tag{2}$$

where $\mathcal{M}_i$ is the set of masked positions and $X_i^{\setminus j}$ denotes the unmasked input with only token at position j masked. At inference, we use the template:

$$X_i' = \texttt{[CLS]}\ X_i\ \texttt{[SEP]}\ \texttt{"The label is"}\ \texttt{[MASK]}\ .$$

and predict the masked label token. The output vocabulary is restricted to the label token set $\mathcal{V}_\mathcal{Y}$.

(3) Auto-regressive modeling (`AR`): Following Radford et al. (2018), we train a causal generative model to minimize the next-token prediction loss over the entire label + input sequence:

$$\mathcal{L}_{\text{gpt}} = -\sum_{i=1}^{N} \sum_{j=1}^{L_i} \log P(x_i^j | y, x_i^1, \dots, x_i^{j-1}) \tag{3}$$

where L_i is the length of the i-th sequence. At inference time, we perform one forward pass per candidate label $y \in \mathcal{V}_\mathcal{Y}$ by prepending it to the input X, and compute the log-likelihood. The predicted label is then obtained as $\arg\max_{y \in \mathcal{V}_\mathcal{Y}} \log P(X \mid y)$. In $\texttt{AR}_{pseudo}$ (refer pseudo-generative models in Section 2) the label is appended at the end instead of the beginning and only one forward pass is required to generate the predicted label token y. Note that label placement is only relevant for causal generative architectures (like `AR`) with a left-to-right attention structure. For bidirectional (pseudo-)generative models like `MLM` or `DIFF`, it has no theoretical impact.

(4) Text Diffusion (`DIFF`): For each input-label pair (X_i, y_i), we first create a template:

$$X_i = X_i\ \texttt{[SEP]}\ \texttt{"The label is"}\ y_i\ .$$

where each template is a sequence $X_i = x_i^1 \dots x_i^{L_i}$ with tokens $x_i^j \in \mathcal{V}$.

Similar to how diffusion models gradually add noise to images, our forward process gradually corrupts text by converting tokens to pure noise (here `[MASK]`). Following Lou et al. (2024), we define the forward process through discrete transition matrices Q_t following a continuous markov process (see eq. 4). This process occurs at different timesteps $t \in [0, T]$, where each token position is independently corrupted, starting from the original text and progressively moving towards a completely masked sequence.

$$\frac{dp_t}{dt} = Q_t\, p_t, \quad \text{with} \quad p_0 = p_{\text{data}} \tag{4}$$

The reverse process learns to reconstruct the original text by predicting what token should replace each `[MASK]` symbol. This is done by learning score ratios $s_\theta(x,t)_z = \frac{p_t(z)}{p_t(x)}$ where x, z are tokens from $\mathcal{V}$ and modeling the reverse process (Sun et al., 2022) as:

$$\frac{dp_{T-t}}{dt} = s_\theta(x,t)_z Q_{T-t}\, p_{T-t} \tag{5}$$

Denoising Score Entropy (DSE) is used for training the score model in a manner that ensures several desired properties for s_θ and ensures the computation is tractable:

$$\mathcal{L}_{\text{DSE}} = \mathop{\mathbb{E}}_{\substack{x_0 \sim p_0, \\ x \sim p(\cdot|x_0)}} \Bigg[\sum_{z \neq x} w_{xz} \Big(s_\theta(x)_z - \frac{p(z \mid x_0)}{p(x \mid x_0)} \log s_\theta(x)_z \Big) \Bigg] \tag{6}$$

where p is assumed to be perturbation of some base density p_0 and weights $w_{xz} > 0$.

The ELBO (Theorem 3.6 in Lou et al. (2024)) provides an upper bound on the negative log-likelihood, which is what we optimize for in generative models:

$$-\log p_0^\theta(x_0) \leq \mathcal{L}_{DWDSE}(x_0) + constant \tag{7}$$

where $\mathcal{L}_{DWDSE}$ integrates $\mathcal{L}_{DSE}$ weighted by the forward diffusion matrix. At inference time, we mask the label token in the template X_i and use the model to predict it, restricting the possible outputs to valid labels in $\mathcal{V}_\mathcal{Y}$. For further details, refer to Lou et al. (2024).

C. Implementation Details

We use the `bert-base-uncased`[1] architecture as the backbone for our **Encoder** and **MLM** experiments, without initializing the model with pretrained weights. This architecture contains approximately 110M parameters, comprising 12 encoder layers, 12 attention heads, and a hidden size of 768. We run all experiments for 3 random seeds and report the average and standard deviation results in main paper.

For the **Encoder** experiments, we conducted a grid search over several hyperparameters, including learning rates of {`1e-5`, `2e-5`, `3e-5`, `4e-5`, `5e-5`}, batch sizes of {32, 64, 128, 256}, and a fixed sequence length of 512 tokens. Training was performed for 30 epochs uniformly for all datasets without early stopping. For the **MLM**-based experiments, we retained similar hyperparameter ranges but trained for 200 epochs to account for the increased complexity of masked token prediction. We observed that adding an early stopping patience parameter sometimes led the model to select a suboptimal checkpoint, as the validation loss often continued to decrease gradually after remaining flat or oscillating for several epochs.

For the **AR** and $\mathbf{AR}_{pseudo}$ experiments, we used the `GPT-2` base architecture[2] as the backbone with 137M parameters comparable with our other experiments. We trained a causal language model to minimize the next-token prediction loss over the concatenated input and label sequence. A grid search was conducted with the same hyperparameter range as mentioned above. The models were trained for up to 100 epochs, with early stopping based on validation loss, using a patience parameter of 10 epochs.

Our **Text Diffusion** approach follows the Diffusion Transformer architecture (Peebles & Xie, 2023) which is basically the vanilla transformer encoder with an extra time-conditioned embedding incorporated with it. The parameter count is $\sim$160M due to the addition of time-dependent embeddings required by the diffusion mechanism. To counter this, we conducted an ablation study by increasing the encoder size to 160M parameters (by adding layers) for other approaches (like `ENC`, `MLM`) to match the diffusion model size, but observed no difference in performance. Hence we retain their original settings as reported above. For diffusion-specific hyperparameters, we used a batch size of 64, learning rate `3e-4` and trained for 200K iterations. We adopted a geometric noise schedule that interpolates between 10^{-4} and 20, similar to the setup in (Lou et al., 2024), and used the following absorbing/masking matrix Q^{absorb} as part of the transition modeling. This was the best hyperparameter setting we found.

$$Q_{\text{absorb}} = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix}$$

All experiments were conducted using multi-GPU training across eight NVIDIA A100 GPUs. Training time varied depending on the methods and configurations used for each dataset. The range of training times (in hours) for various datasets is presented in Table 1. All reported training times correspond to full-data training configurations.

D. Dataset Details

AG News (Zhang et al., 2015): It consists of approximately 120K training samples and 7.6K test samples, divided into four categories: World, Sports, Business, and Technology. Each sample contains a short news article, typically consisting of the title and the first few sentences. **Emotion** (Saravia et al., 2018): A collection of English tweets labeled with six basic emotions: anger, fear, joy, love, sadness, and surprise. It is designed for emotion detection in text. The dataset has 20K samples divided into 16K samples for training and 2K samples each for validation and testing. **Stanford Sentiment**

[1] https://huggingface.co/google-bert/bert-base-uncased
[2] https://huggingface.co/openai-community/gpt2

Config	ENC	AR_{pseudo}	AR	MLM	DIFF
(1L,1H)	1-2	2-4	2-4	1-4	1-4
(6L,6H)	1-3	3-7	3-7	3-7	2-6
(12L,12H)	2-5	5-10	5-10	5-10	5-12

Table 1: Training time (in hrs) ranges across different datasets for each configuration and approach.

Dataset	Split	Examples	Classes	Avg Tokens	Label Dist. (%)	Ordinal
IMDb	train	25,000	2	313.87	0: 50.0, 1: 50.0	×
	test	25,000	2	306.77	0: 50.0, 1: 50.0	
agnews	train	120,000	4	53.17	0-3: 25.0 each	×
	test	7,600	4	52.75	0-3: 25.0 each	
emotion	train	16,000	6	22.26	0: 29.2, 1: 33.5, 2: 8.2, 3: 13.5, 4: 12.1, 5: 3.6	×
	test	2,000	6	21.90	0: 27.5, 1: 35.2, 2: 8.9, 3: 13.8, 4: 10.6, 5: 4.1	
hatespeech	train	22,783	3	30.04	0: 5.8, 1: 77.5, 2: 16.7	✓
	test	2,000	3	30.18	0: 5.5, 1: 76.6, 2: 17.9	
multiclasssentiment	train	31,232	3	26.59	0: 29.2, 1: 37.3, 2: 33.6	✓
	test	5,205	3	26.91	0: 29.2, 1: 37.0, 2: 33.8	
rottentomatoes	train	8,530	2	27.37	0: 50.0, 1: 50.0	×
	test	1,066	2	27.32	0: 50.0, 1: 50.0	
sst2	train	6,920	2	25.21	0: 47.8, 1: 52.2	×
	test	872	2	25.47	0: 49.1, 1: 50.9	
sst5	train	8,544	5	25.04	0: 12.8, 1: 26.0, 2: 19.0, 3: 27.2, 4: 15.1	✓
	test	1,101	5	25.24	0: 12.6, 1: 26.3, 2: 20.8, 3: 25.3, 4: 15.0	
twitter	train	9,543	3	27.62	0: 15.1, 1: 20.2, 2: 64.7	✓
	test	2,388	3	27.92	0: 14.5, 1: 19.9, 2: 65.6	

Table 2: Dataset statistics showing training and test split sizes, number of classes, mean and maximum token lengths, and label distribution percentages. Refer to Section D for details on datasets.

Treebank (SST) (Socher et al., 2013): We utilize both the SST-2 (binary sentiment) and SST-5 (fine-grained sentiment) variants of the Stanford Sentiment Treebank dataset. SST-2 consists of sentences labeled as either positive or negative, suitable for binary sentiment classification, while SST-5 includes five sentiment categories: very negative, negative, neutral, positive, and very positive, allowing for more fine-grained sentiment analysis. **Multiclass Sentiment Analysis** [3]: This dataset consists of 41.6K data points, labeled into three sentiment categories: positive, negative, and neutral. While the dataset is designed for multiclass sentiment classification, it exhibits class imbalance, with certain sentiment classes being more prevalent than others. This imbalance provides a more realistic challenge for sentiment analysis models, testing their ability to handle skewed distributions and still perform effectively across all sentiment categories. **Twitter Financial News Sentiment** [4]: A specialized English-language collection of finance-related tweets, annotated for sentiment analysis. It consists of 11,932 tweets labeled with three sentiment categories: Bearish, Bullish, and Neutral. This dataset is designed to test models' ability to understand domain-specific language and nuanced sentiment expressions in financial contexts. **IMDb** (Maas et al., 2011): A binary sentiment analysis dataset consisting of 50K reviews from the Internet Movie Database (IMDb), labeled as positive or negative. The dataset is balanced, with an equal number of positive and negative reviews. This dataset is characterized by longer document lengths and detailed opinions, making it a challenging benchmark. **Rotten Tomatoes** (Pang & Lee, 2005): A binary classification dataset which contains 10,662 movie review sentences, equally

[3]https://huggingface.co/datasets/Sp1786/multiclass-sentiment-analysis-dataset

[4]https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

divided into 5,331 positive and 5,331 negative examples. The dataset is characterized by relatively short, opinion-driven sentences that reflect concise sentiments about films. **Hate Speech Offensive** (Davidson et al., 2017): A major challenge in automatic hate speech detection is distinguishing hate speech from other forms of offensive language. This dataset consists of approximately 25K tweets, labeled into three categories: hate speech, offensive language without hate speech, and neutral content.

Refer to Table 2 for details on dataset statistics.

E. More Ordinal & Calibration Results

In this section, we take a closer look at ordinal and calibration results for the datasets decribed above. Here we report ordinal metrics on the datasets **Stanford Sentiment Treebank (SST5)** (Socher et al., 2013), **Multiclass Sentiment Analysis**, **Hate Speech Offensive** (Davidson et al., 2017) and **Twitter Financial News Sentiment** since these are the only multi-class ordinal datasets out of 9. Calibration metrics are reported on all 9 datasets.

In Figure 4, we compare how ordinal and calibration metrics vary with increasing model size. Figure 5 presents the ordinal metrics for all four ordinal datasets, while Figure 6 shows the calibration metrics for all nine datasets. The corresponding insights are discussed in Section 4.

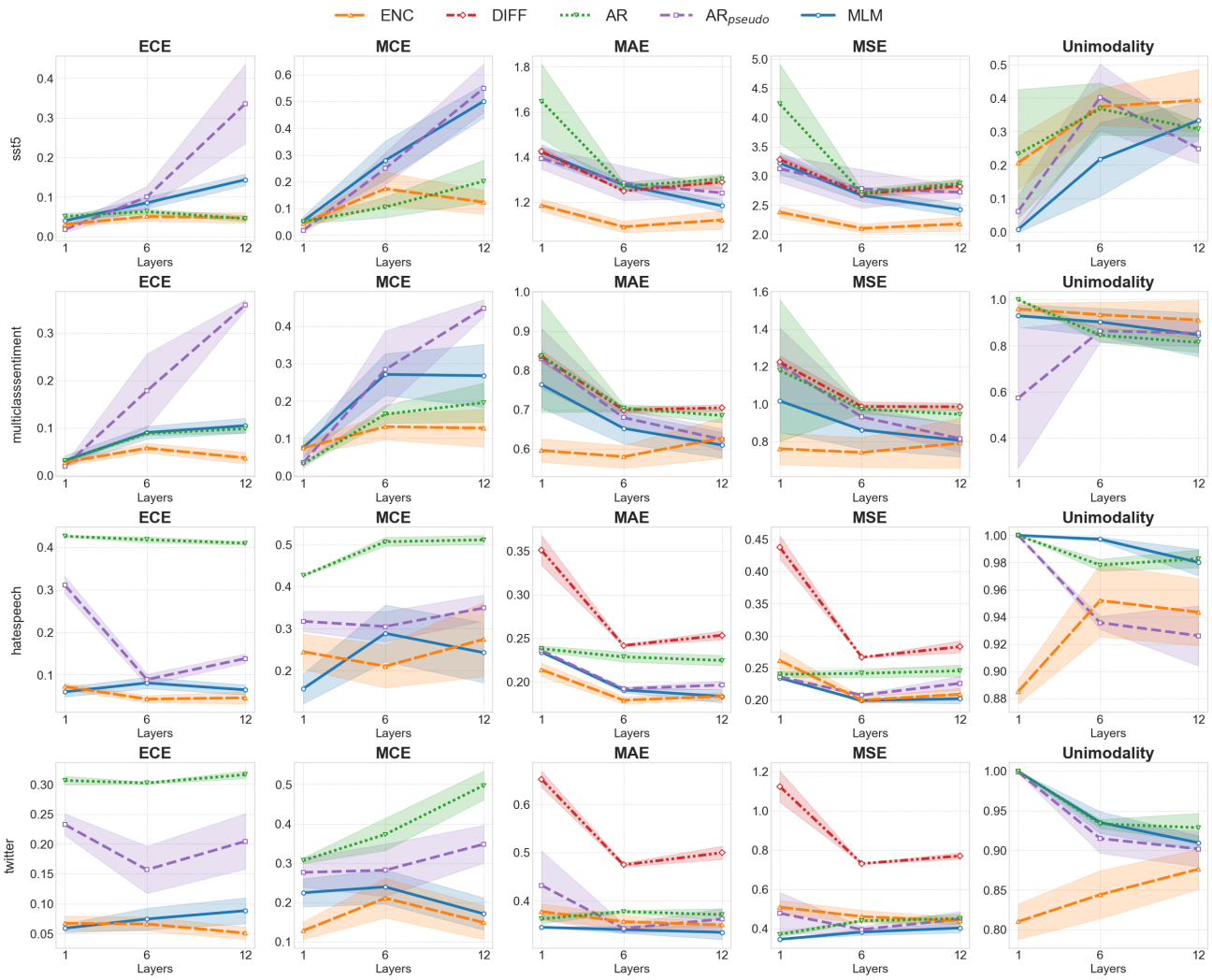

Figure 4: **[Best viewed in color]** Calibration and Ordinal metrics comparison across layers 1, 6 and 12. For ECE, MCE, MAE, MSE, (↓ is better) and UM (↑ is better).

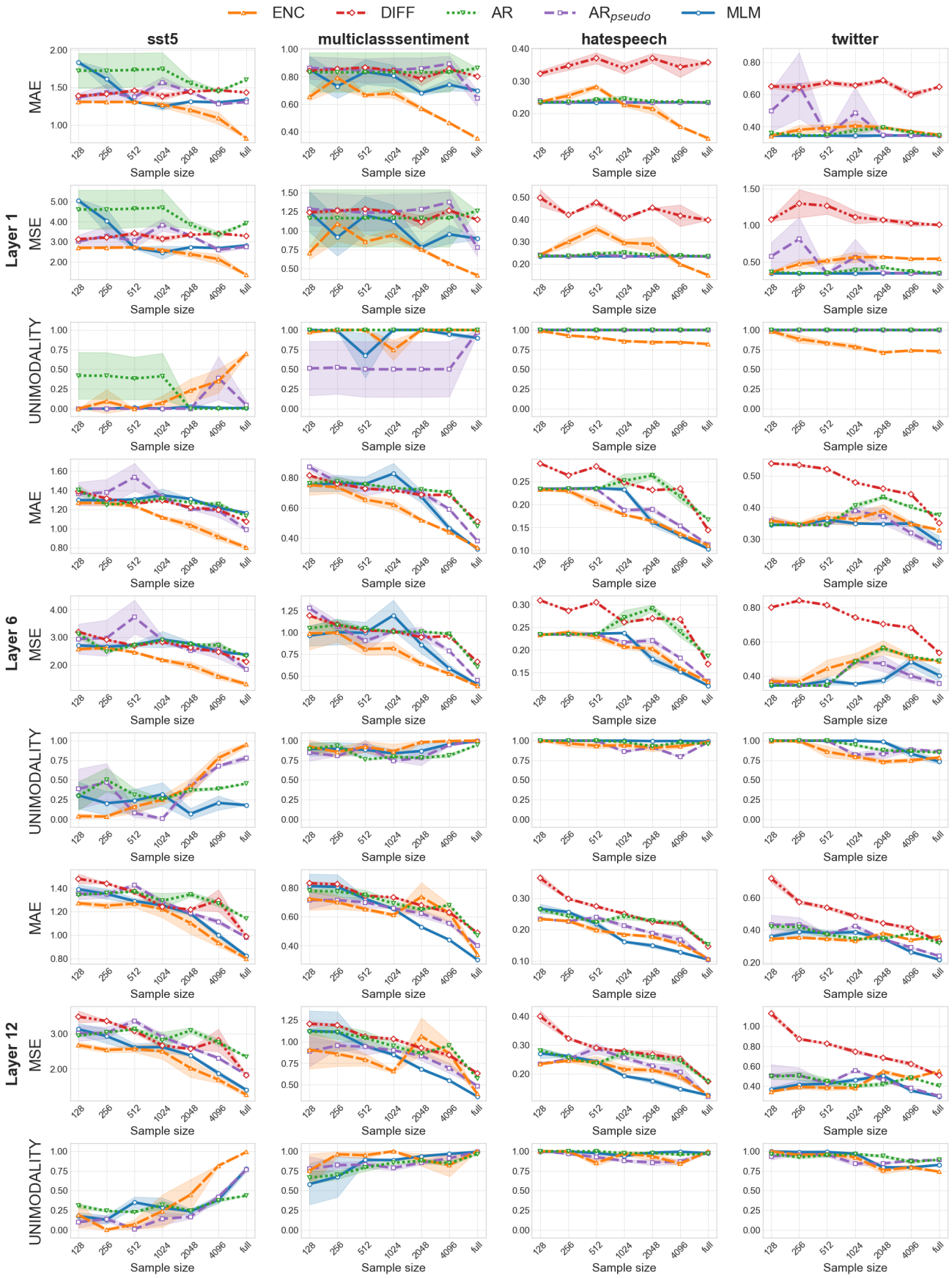

Figure 5: **[Best viewed in color]** Ordinal metrics. For MAE, MSE, (↓ is better) and UM (↑ is better).

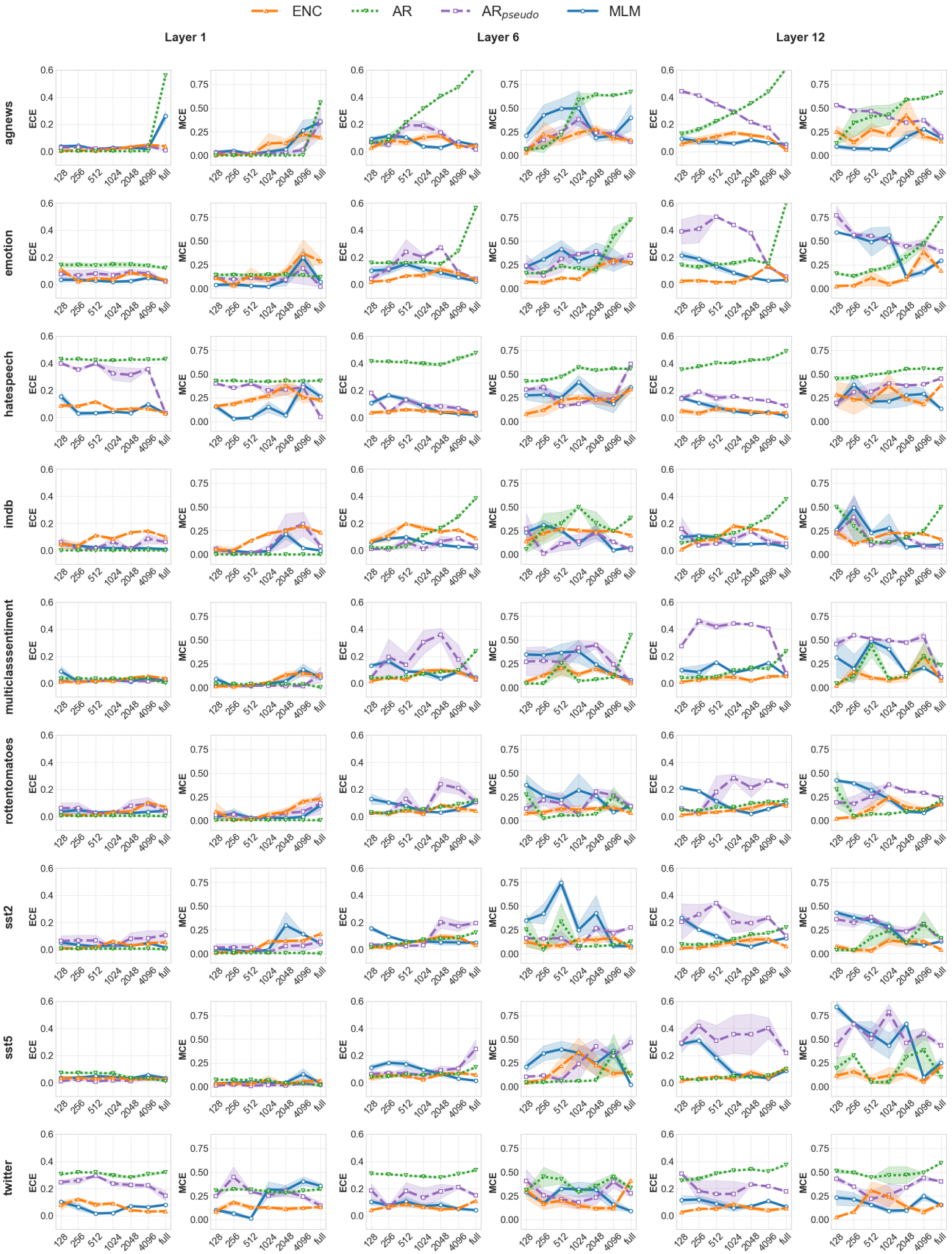

Figure 6: **[Best viewed in color]** Calibration metrics. For ECE, MCE (↓ is better)

F. More Main Results

This section contains the extended results of Figure 2 (see Figure 7) and Figure 3 (see Figure 8) for all 9 datasets. We omit 1-layer plots for Figure 8 since the performance is mostly trivial for low-data settings and the same trend is observed as 6/12-layers for full-data settings.

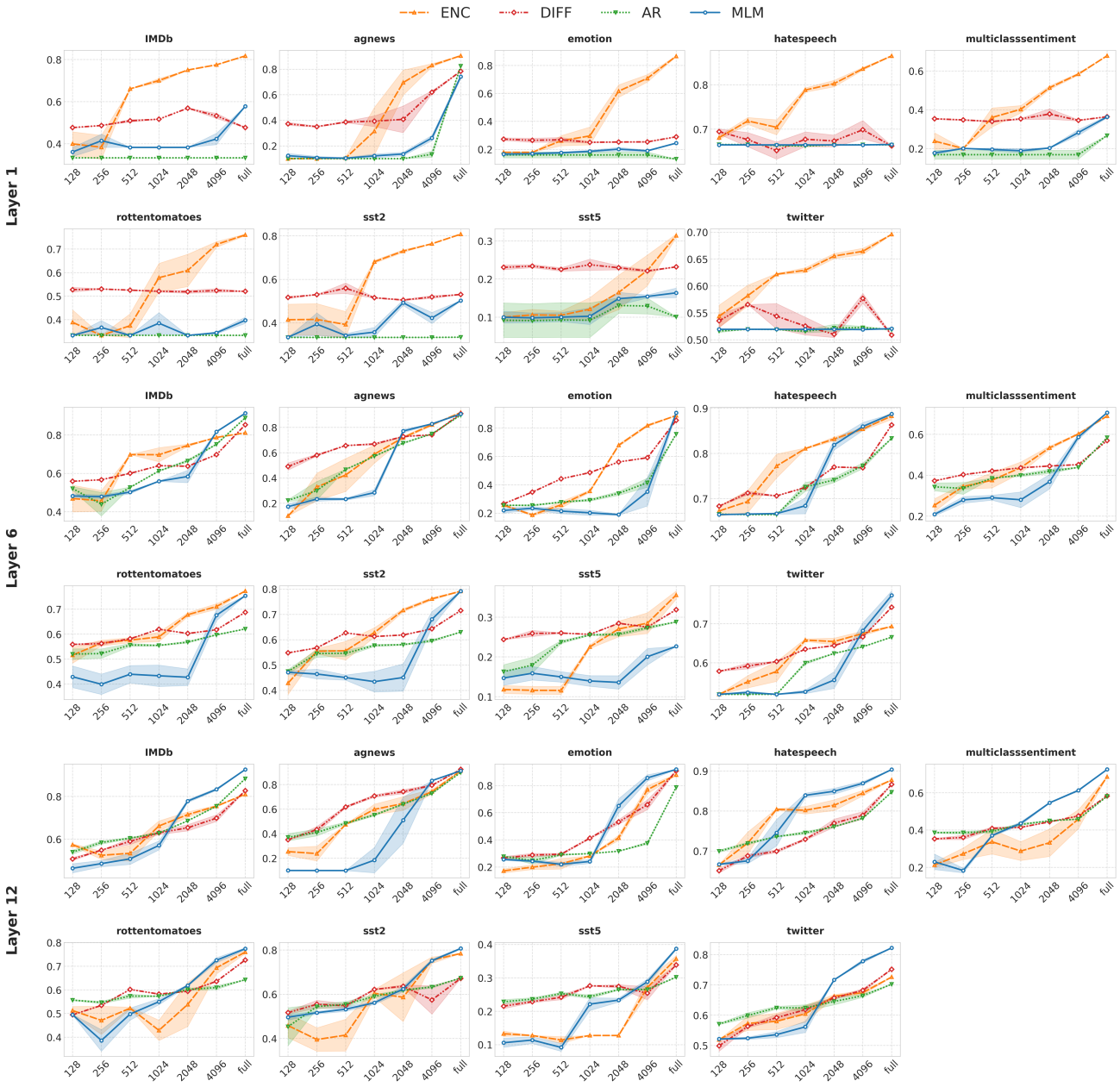

Figure 7: **[Best viewed in color]** Comparison of weighted-F1 scores of models across different configurations for all 9 datasets. (↑ is better) (`X-axis`: sample size, `Y-axis`: weighted-F1 score)

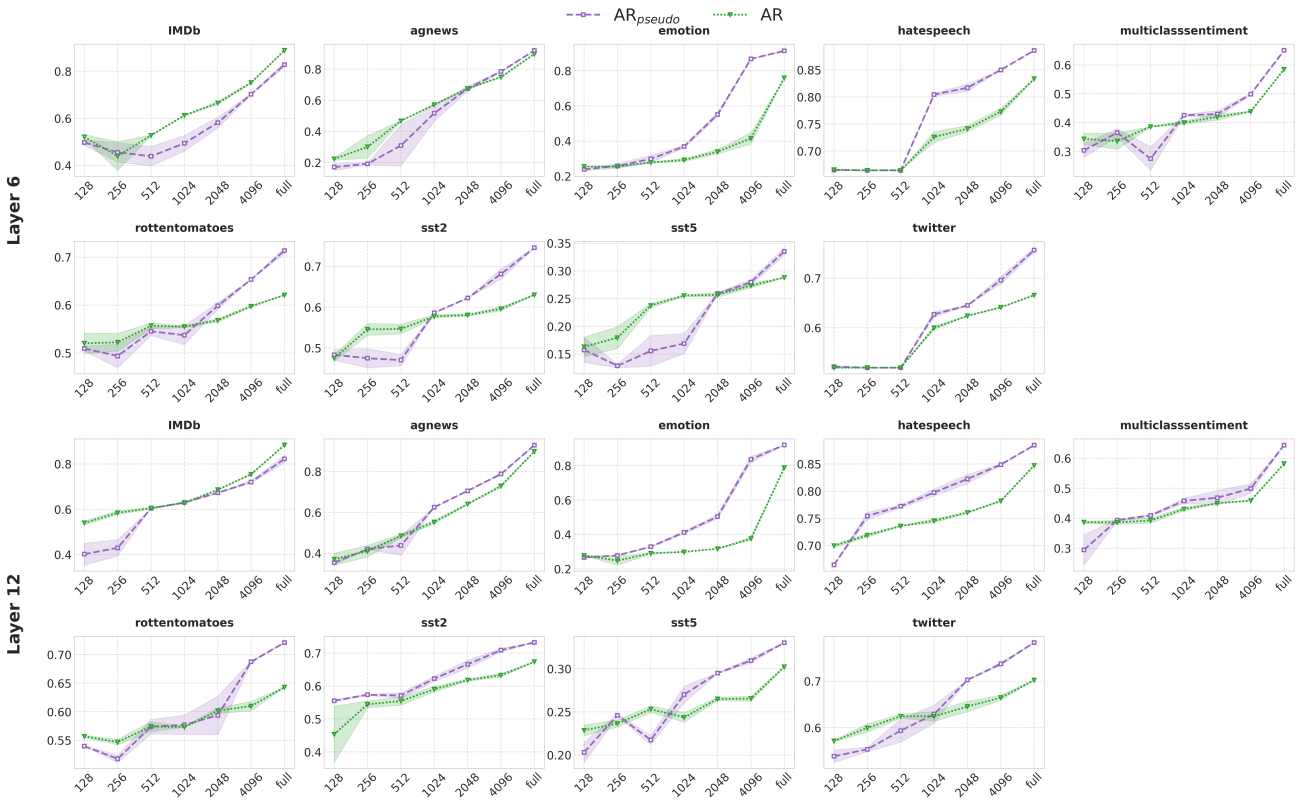

Figure 8: **[Best viewed in color]** Comparison of weighted-F1 scores between $\texttt{AR}_{pseudo}$ and AR (↑ is better) for all datasets. (`X-axis`: sample size, `Y-axis`: weighted-F1 score)