Domain Gating Ensemble Networks for AI-Generated Text Detection

Arihant Tripathi[*], **Liam Dugan**[*], **Charis Gao**, **Maggie Huan**,
Emma Jin, **Peter Zhang**, **David Zhang**, **Julia Zhao**, **Chris Callison-Burch**
University of Pennsylvania
{atrip, ldugan, ccb}@seas.upenn.edu

Abstract

As state-of-the-art language models continue to improve, the need for robust detection of machine-generated text becomes increasingly critical. However, current state-of-the-art machine text detectors struggle to adapt to new unseen domains and generative models. In this paper we present DoGEN (Domain Gating Ensemble Networks), a technique that allows detectors to adapt to unseen domains by ensembling a set of domain expert detector models using weights from a domain classifier. We test DoGEN on a wide variety of domains from leading benchmarks and find that it achieves state-of-the-art performance on in-domain detection while outperforming models twice its size on out-of-domain detection. We release our code and trained models[1] to assist in future research in domain-adaptive AI detection.

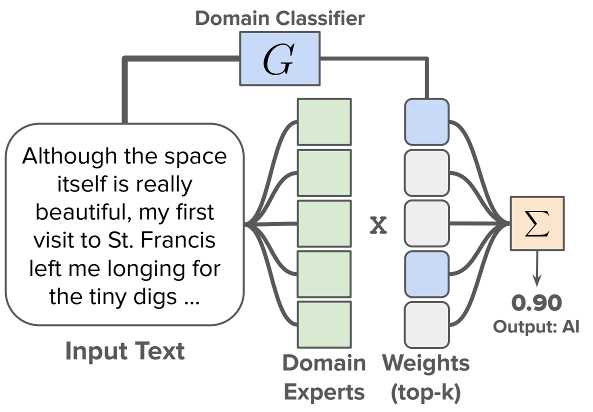
Domain Classifier
G
Although the space itself is really beautiful, my first visit to St. Francis left me longing for the tiny digs ...
Input Text
x
Domain Experts
Weights (top-k)
Σ
0.90
Output: AI

Figure 1: The Domain Gating network splits the input into a probability distribution over N experts. The output is a weighted sum of the outputs of the top k experts.

1 Introduction

Large Language Models (LLMs) are extremely good at generating human-like text. Such text is frequently mistaken for human-written text by non-expert readers and scholars alike (Dugan et al., 2020, 2023; Clark et al., 2021). Therefore, robust and cheap AI detection technology would help significantly reduce the plethora of negative externalities of LLMs such as scientific fraud (Lund et al., 2023), targeted phishing scams (Hazell, 2023), and disinformation (Spitale et al., 2023).

Recent work in AI text detection has largely focused on training single models on a variety of domains and generators (Emi and Spero, 2024; Hu et al., 2023). However, these models have been shown to lack robustness to unseen domains and generators (Dugan et al., 2024).

To address these limitations, we propose DoGEN (Domain Gating Ensemble Networks). We first use a *domain router* network to output a probability distribution over N domains for a given input document. This distribution is then used to weight an ensemble of domain-specific expert detectors in order to predict the correct output class (Figure 1). In our experiments we show that DoGEN achieves state-of-the-art performance for in-domain detection and also outperforms other similar ensembling techniques on unseen domains and generators.

2 Related Work

Neural Ensemble Methods There have been many attempts to ensemble neural classifiers to detect machine-generated text. The most common approach collects zero-to-one scores from various classifiers and trains a lightweight classification layer on top to learn weights for the different classifiers (Abburi et al., 2023; Lai et al., 2024; Liyanage and Buscaldi, 2023; Nguyen et al., 2023; Joy and Aishi, 2023). While this technique works well on in-domain data, it fails to generalize well to generators and domains that are not seen during training.

Metric-Based Ensembles Another related line of work is ensembling features not derived from

[*]Equal contribution
[1]`https://github.com/Siris2314/dogen`

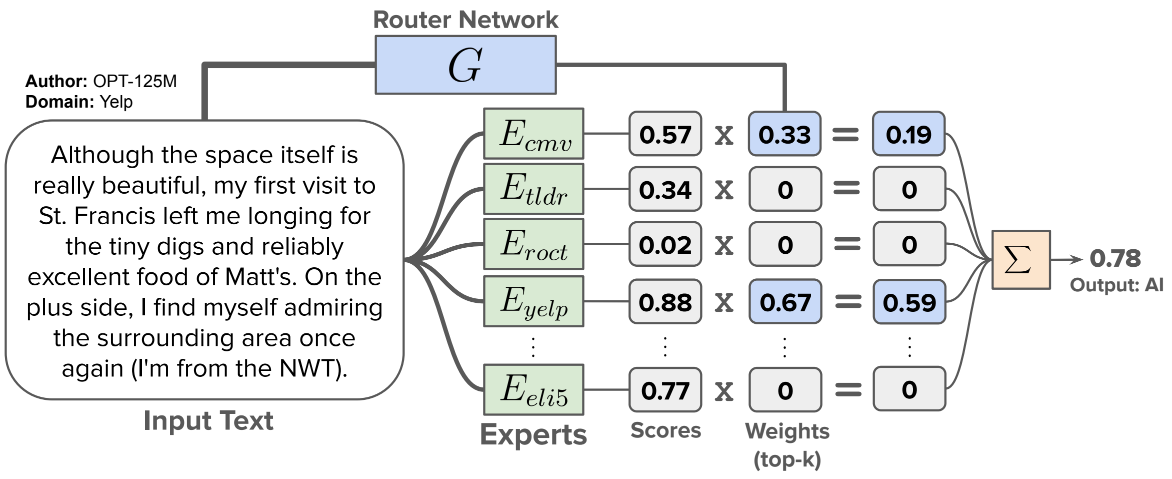
Router Network
G
Author: OPT-125M
Domain: Yelp
Although the space itself is really beautiful, my first visit to St. Francis left me longing for the tiny digs and reliably excellent food of Matt's. On the plus side, I find myself admiring the surrounding area once again (I'm from the NWT).
Input Text
E_{cmv} 0.57 x 0.33 = 0.19
E_{tldr} 0.34 x 0 = 0
E_{roct} 0.02 x 0 = 0
E_{yelp} 0.88 x 0.67 = 0.59
E_{eli5} 0.77 x 0 = 0
$\sum$
0.78
Output: AI
Experts
Scores
Weights (top-k)

Figure 2: A graphical representation of the full Domain Gating Ensemble Network (DoGEN). Texts are given to each of N domain-specific experts. Scores from each expert are ensembled using weights determined by the *router network*. This network is trained for domain classification and the top-k experts are chosen for output.

trained classifiers but rather derived from linguistic stylometry or logits of other LLMs. Verma et al. (2024) learns an ensemble by conducting a structured search over combinations of such features. Ong and Quek (2024) ensemble the negative curvature of log likelihood from many different LLMs. Finally, Mao et al. (2025) learn ensemble weights over log-rank and entropy from many different LLMs. While these techniques are effective at adapting to out-of-domain settings, they struggle to achieve high performance on in-domain text as their underlying features are generic and not optimized for any particular generator or domain.

Domain-Adaptation Techniques Recent work has begun to incorporate techniques for learning domain-agnostic features for AI-text detection (Agrahari et al., 2025). Bhattacharjee et al. (2023, 2024) use Gradient Reversal Layers (GRL) along with a reconstruction loss. Abassy et al. (2024) use the same technique along with a multi-class classification objective. Gui et al. (2025) additionally learn a Variational Information Bottleneck (VIB) to assist in reconstruction. These techniques have shown a lot of promise, however, there are many valuable domain-specific features that one may want to use for detection. Our proposed method learns when—and when not—to use such features allowing for a more flexible approach.

3 Our Approach

Let $\mathcal{X}$ denote the space of input documents and $\mathcal{Y}$ the space of (scalar) detector scores we wish to predict.[2] We fine-tune N specialists ("experts")

$$E_i : \mathcal{X} \longrightarrow \mathcal{Y}, \qquad i \in \{1, \ldots, N\},$$

each optimized on a single domain $\mathcal{D}_i \subset \mathcal{X}$. We then fine tune a router for domain classification $G : \mathcal{X} \to \Delta^{N-1}$ that maps each document to a probability distribution over the N domains $G(x; \phi, W) = (p_1, \ldots, p_N)$,

$$p_i(x) = \frac{\exp\left(\mathbf{w}_i^\top \phi(x)\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{w}_j^\top \phi(x)\right)},$$

where $\phi(x) \in \mathbb{R}^d$ is an encoder representation and $\{\mathbf{w}_i\}_{i=1}^N$ are learnable parameters. During *router training* we freeze the experts and minimize a standard cross-entropy loss

$$\mathcal{L}_{\text{gate}} = -\frac{1}{M} \sum_{m=1}^{M} \log p_{d_m}\left(x^{(m)}\right),$$

where d_m is the oracle domain label for sample $x^{(m)}$. This loss encourages G to give high likelihood to the domain–appropriate expert. Both the classification head $\{\mathbf{w}_i\}_{i=1}^N$ and the encoder ϕ are trained end-to-end.

Prediction rule. Given $p_1, \ldots, p_N$ and the experts' raw scores $\mathbf{y}(x) = E_1(x), \ldots, E_N(x)$ we compute the final detector score with the following strategy: Let $\mathcal{I}_k$ be the indices of the k largest $p_i(x)$, and softmax $w_i = p_i / \sum_{j \in \mathcal{I}_k} p_j$ for $i \in \mathcal{I}_k$.

$$s^{(k)}(x) = \sum_{i \in \mathcal{I}_k} w_i \, E_i(x), \;\; k \in \{1, \ldots, N\}$$

Setting $k = N$ recovers a full dot-product gating.

[2]In our setting $\mathcal{Y} = [0, 1]$, where values closer to 1 indicate a higher probability that the text was machine-generated.

4 Experiments

4.1 Data

For our training data, we use the train split of **MAGE** (Li et al., 2024), a large corpus designed for training and evaluating machine-generated text detectors. The training set of MAGE contains text from 10 domains: `cmv`, `eli5`, `tldr`, `xsum`, `wp`, `roct`, `hswag`, `yelp`, `squad`, and `sci_gen`. To address class imbalance, we down-sample the data to ensure an equal number of machine-generated and human-written documents per domain[3]. After down-sampling, each domain is split 90:10 into training and validation sets.

To test our models we use the test set of **MAGE** which includes data from these same 10 domains as well as 4 additional domains: `cnn`, `dialog_sum`, `imdb`, and `pubmed`—which are unseen during training. In addition, we use the test set of the **RAID** benchmark (Dugan et al., 2024), focusing only on its non-adversarial subset to evaluate generalization across models, domains, and decoding strategies.

4.2 Metrics

For evaluation we use AUROC. This metric measures the probability that a detector ranks a randomly chosen machine-generated document higher than a human-written one. Because it integrates over all possible thresholds, AUROC is threshold-free, insensitive to class imbalance, and reflects overall ranking quality.

4.3 Expert and Router Models (DoGEN)

For DoGEN we fine-tune $N = 10$ expert models for AI-text detection, one on each domain in the MAGE training set. Each expert uses a Qwen1.5-1.8B[4] base model with a binary classification head.

For the router network we again fine-tune Qwen1.5-1.8B to classify documents into the 10 domains in the training set of MAGE. At inference time we employ a top-k routing strategy with $k = 2$, meaning that each input is routed to the top two experts based on the router's selection scores.

We fine-tune all models using the HuggingFace `Trainer` API. For full training details and hyperparameter settings, refer to Appendix A.1.

4.4 Comparisons

We compare DoGEN against the following baselines, all fine-tuned with the same hyper-parameters (Table 3) and class-balancing strategy:

Qwen1.5 As an initial baseline we trained two base models, Qwen1.5-1.8B and Qwen1.5-32B[5] on all domains of MAGE. The 1.8B model was chosen to match the parameter count of a single expert and the 32B model was chosen to roughly match the parameter count of the full ensemble. This serves as a direct comparison to our experts, which all are fine-tuned versions of Qwen1.5. In essence, this helps us to understand how well an expert that was trained on all of MAGE would do on this task.

Qwen1.5-MoE-A2.7B This model adopts a Mixture-of-Experts (MoE) architecture by replacing the feed-forward layers in Qwen1.5-1.8B with MoE layers (Qwen Team, 2024). It has 14.3B parameters in total, with only 2.7B activated during runtime. We trained this model on all domains of MAGE and chose it model to highlight the specific advantage of our domain-aware gating strategy relative to a Mixture-of-Experts design.

Equal Vote This comparison takes the trained experts and ensembles them together using an equal voting strategy. More specifically, given the experts' raw scores $\mathbf{y}(x) = E_1(x), \ldots, E_N(x)$ we compute the final detector score as a weighted sum where all ensemble weights are $1/N$: $\frac{1}{N}\sum_{j=1}^{N} E_j$.

Weighted Vote This comparison follows Abburi et al. (2023) and uses Logistic Regression to learn ensemble weights for the N experts. Unlike DoGEN, these weights are static and do not change based on the input text. The exact learned weights can be found in Appendix A.4.

Joint Training This comparison takes both the router network G and experts $E_1 \cdots E_N$ and trains the full ensemble end-to-end on all domains. We conduct this experiment with two different initialization mechanisms. The first is initialized from the pre-trained Qwen1.5-1.8B model (JT-Scratch) and the second is initialized from the existing DoGEN model checkpoints (JT-Domain). Training was done with $k = N$ routing (all experts active) and evaluation was done with $k = 2$ routing. We chose these comparisons to see whether the full ensemble would learn domain routing when trained end-to-end.

[3] see Appendix A.2 for ablations

[4] `https://huggingface.co/Qwen/Qwen1.5-1.8B`

[5] `https://huggingface.co/Qwen/Qwen1.5-32B`

MAGE – AUROC (%)														
Model	cmv	eli5	tldr	xsum	wp	roct	hswag	yelp	squad	s_gen	cnn	d_sum	imdb	pubmed	all
Qwen1.8B	**99.89**	98.33	98.49	**99.71**	98.97	98.89	98.89	**98.69**	97.79	97.74	71.58	39.05	91.56	81.97	88.41
Qwen32B	98.52	96.22	94.71	96.60	96.62	91.06	91.03	95.15	97.91	97.94	**93.52**	**90.50**	93.89	**90.97**	95.20
QwenMoE	98.60	95.91	96.35	90.10	98.87	94.78	94.97	95.17	98.03	94.08	86.53	75.11	**96.08**	87.66	95.43
Equal Vt.	95.06	97.30	98.14	95.63	99.38	97.01	95.64	96.27	98.82	98.25	53.79	38.04	86.01	84.24	95.88
Weight Vt.	99.04	97.06	98.08	98.17	99.48	98.30	98.10	95.63	98.59	97.19	65.43	39.40	82.32	80.11	91.71
JT-Domain	99.74	98.50	**98.99**	99.09	99.56	**98.92**	98.08	98.20	**99.43**	**99.28**	61.44	62.49	81.62	77.38	96.74
JT-Scratch	98.91	96.51	96.09	96.21	99.20	91.76	89.30	94.68	98.01	98.56	76.59	62.28	81.23	80.26	94.17
DoGEN	99.73	**98.76**	**98.99**	99.01	**99.76**	98.76	**99.04**	97.81	99.06	99.15	78.98	45.64	84.93	79.10	**97.60**

Table 1: Performance on detecting machine-generated text in various domains from the MAGE benchmark. Bold values denote the best score in each column. We see that DoGEN performs the best overall on *in-domain* data.

RAID – AUROC (%)									
Model	abstracts	books	news	poetry	recipes	reddit	reviews	wiki	all
Qwen1.8B	92.95	94.93	95.95	91.18	84.30	91.33	96.52	90.64	93.34
Qwen32B	94.09	97.10	95.72	94.42	96.64	92.68	97.41	93.08	94.73
QwenMoE	88.24	91.99	92.81	92.75	82.22	93.75	94.72	95.75	91.34
Equal Vote	95.51	**98.47**	96.54	93.92	83.68	**96.85**	**98.02**	94.72	92.73
Weighted Vote	92.31	95.51	94.57	89.35	90.04	90.59	93.01	94.48	92.03
JT-Domain	**96.84**	96.73	97.86	90.85	69.89	94.73	97.10	96.34	92.27
JT-Scratch	95.36	97.10	**98.51**	**98.08**	87.49	94.57	94.47	96.37	95.65
DoGEN	95.91	97.22	98.09	91.05	**98.05**	95.08	97.22	**96.71**	**95.81**

Table 2: Performance of various models on detecting machine-generated text on the RAID benchmark. Bold values denote the best score in each column. We see that DoGEN performs the best overall on *out-of-domain* data.

5 Results

In-domain (MAGE). Table 1 reports performance on in-domain text from the MAGE benchmark. DoGEN achieves the highest overall AUROC (**97.60**%), outperforming both the 32B dense baseline (95.20%) and the off-the-shelf Mixture-of-Experts model (95.43%). While the larger Qwen1.5-32B model performs well, it cannot specialize as effectively as our ensemble, which selects an expert fine-tuned for each specific domain.

At inference time, DoGEN activates only two 1.8B experts plus the gating network (5.4B parameters total), offering a parameter-efficient alternative to a 32B dense model. While the ensemble shows strong in-domain performance, it underperforms on certain out-of-distribution domains such as dialog_sum (see Appendix D.2 for discussion). Nonetheless, DoGEN's modular structure allows us to add additional experts for targeted adaptation without retraining the entire model.

Out-of-domain (RAID). Table 2 evaluates the same models on out-of-domain text from the RAID benchmark. We see that DoGEN again achieves the highest AUROC (95.81%), outperforming both the 32B model (94.73%) and the MoE baseline (91.43%). Surprisingly, although the ensemble was designed for domain-specific specialization, it is able to generalize comparatively well to out-of-distribution examples. One possible explanation is that, on new unseen domains, the router network is able to choose experts that specialize in similar enough domains to still be able to predict accurately. We investigate this further in Appendix C.

6 Conclusion

In the real-world, AI-generated text detection will mainly be used in a limited selection of high-risk domains (e.g. student essays, academic papers, news articles, etc.). In order to accurately detect text from these domains while remaining competitive in other rarer domains, detectors must be able to decide what models to use depending on the characteristics of the input text.

In this paper we propose a method to accomplish this by ensembling together a set of expert detectors via a gating network trained for domain classification. We show that not only do we achieve high accuracy on domains the model has seen during training, we also get very high performance on new, unseen domains—demonstrating the flexibility of the technique.

Limitations

One limitation of this paper is the lack of diversity of base models used in testing due to resource constraints. Our domain router and experts both use Qwen1.5-1.8B as a base and we do not test other potential models. Such experiments would allow us to rule out the possibility that the specific pre-training data distribution used to train Qwen can account for our positive result.

While we do test on many OOD domains and models, there are many other possible evaluation datasets that we did not choose to use such as SemEval (Wang et al., 2024a), M4 (Wang et al., 2024b), MULTITuDE (Macko et al., 2023), etc. Future work should seek to incorporate multilingual text into this method—training a multilingual ensemble of experts with a language ID classification head is a promising future direction.

Acknowledgements

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, Jonibek Mansurov, Ekaterina Artemova, Vladislav Mikhailov, Rui Xing, Jiahui Geng, Hasan Iqbal, Zain Muhammad Mujahid, Tarek Mahmoud, Akim Tsvigun, and 5 others. 2024. LLM-DetectAIve: a tool for fine-grained machine-generated text detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 336–343, Miami, Florida, USA. Association for Computational Linguistics.

Harika Abburi, Kalyani Roy, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. A simple yet efficient ensemble approach for AI-generated text detection. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 413–421, Singapore. Association for Computational Linguistics.

Shifali Agrahari, Prabhat Mishra, and Sujit Kumar. 2025. Random at GenAI detection task 3: A hybrid approach to cross-domain detection of machine-generated text with adversarial attack mitigation. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 365–370, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *Preprint*, arXiv:2403.15690.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Tao Yu, Felipe Such, Xin Li, Jacob Austin, Andrew M. Dai, Adam Roberts, Francois Chollet, and Quoc V. Le. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

NLP Team Cohere. 2024. World-class ai, at your command. Accessed: 2024-02-02.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A tool for evaluating human detection of machine-generated text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 189–196, Online. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text? investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Bradley Emi and Max Spero. 2024. Technical report on the pangram ai-generated text classifier. *Preprint*, arXiv:2402.14873.

Jiayi Gui, Baitong Cui, Xiaolian Guo, Ke Yu, and Xiaofei Wu. 2025. AIDER: a robust and topic-independent framework for detecting AI-generated text. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9299–9310, Abu Dhabi, UAE. Association for Computational Linguistics.

Julian Hazell. 2023. Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Preprint*, arXiv:2307.03838.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Saman Sarker Joy and Tanusree Das Aishi. 2023. Feature-level ensemble learning for robust synthetic text detection with DeBERTaV3 and XLM-RoBERTa. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pages 169–172, Melbourne, Australia. Association for Computational Linguistics.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024. Adaptive ensembles of fine-tuned transformers for llm-generated text detection. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Vijini Liyanage and Davide Buscaldi. 2023. An ensemble method based on the combination of transformers with convolutional neural networks to detect artificially generated text. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pages 107–111, Melbourne, Australia. Association for Computational Linguistics.

Brady D Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, Somipam Shimray, and Ziang Wang. 2023. Chatgpt and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74(5):570–581.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Dianhui Mao, Denghui Zhang, Ao Zhang, and Zhihua Zhao. 2025. Mlsdet: Multi-llm statistical deep ensemble for chinese ai-generated text detection. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

NLP Team MosaicML. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

Duke Nguyen, Khaing Myat Noe Naing, and Aditya Joshi. 2023. Stacking the odds: Transformer-based ensemble for AI-generated text detection. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pages 173–178, Melbourne, Australia. Association for Computational Linguistics.

Ivan Ong and Boon King Quek. 2024. Applying ensemble methods to model-agnostic machine-generated text detection. *Preprint*, arXiv:2406.12570.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Qwen Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Vemuri, and 15 others. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations (ICLR)*.

Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. 2023. Ai model gpt-3 (dis)informs us better than humans. *Science Advances*, 9(26):eadh1850.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Haiyang Xu, Xiao Cheng, Shaohan Huang, Wenhui Wang, Furu Wei, Huadong Chen, Lidong Zhou, Zhifang Sui, Daxin Jiang, and Ming Zhou. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Hyper-parameter	Value
per_device_train_batch_size	8
per_device_eval_batch_size	8
num_train_epochs	3
learning_rate	5×10^{-6}
evaluation_strategy	steps
eval_steps	100
early_stopping_patience	10
logging_steps	100
dataloader_num_workers	2

Table 3: Hyper-parameters used in all experiments.

A Training Details

A.1 Hyperparameters & Configurations

Hyperparameters Table 3 lists the full set of hyper-parameters used for all experiments reported in this paper. The experts and domain routing network were trained using the Huggingface Trainer module while the Joint Training experiments were conducted using PyTorch.

A.2 Training Data Pre-processing

The distribution of human and AI-generated examples in the MAGE training data is highly imbalanced across domains prior to any preprocessing. For example, `roct` includes 25,510 AI examples but only 3,287 human examples, while `yelp` contains 31,827 human examples compared to 20,388 AI. In Table 5 we report the original counts as well as the balanced counts used during training.

MAGE – AUROC (%)															
Strategy	`cmv`	`eli5`	`tldr`	`xsum`	`wp`	`roct`	`hswag`	`yelp`	`squad`	`scigen`	`cnn`	`d_sum`	`imdb`	`pubmed`	**All**
Per-domain	**99.89**	**98.33**	**98.49**	**99.71**	98.97	**98.89**	**98.89**	**98.69**	**97.79**	**97.74**	**71.58**	**39.05**	**91.56**	**81.97**	**88.41**
Global	91.77	80.41	75.76	79.78	92.51	75.24	56.01	40.64	65.71	76.01	25.83	38.17	79.33	60.00	71.98
Unbalanced	99.05	89.51	91.44	90.26	**99.55**	82.24	84.16	88.38	93.23	96.76	24.17	29.83	41.50	45.00	86.73

Table 4: AUROC (%) on MAGE-test of a single Qwen1.5-1.8B classifier trained on all documents in MAGE-train with various balancing strategies. **Per-domain**: per-domain training with human/AI balance. **Global**: Global balance across all domains. **Unbalanced**: No balancing done, Bold denotes the best score per column.

Domain	**MAGE (Hum. / AI)**	**Balanced (Hum. / AI)**
`cmv`	4,223 / 20,388	4,223 / 4,223
`eli5`	16,706 / 25,548	16,706 / 16,706
`hswag`	3,129 / 24,482	3,129 / 3,129
`roct`	3,287 / 25,510	3,287 / 3,287
`sci_gen`	4,436 / 18,691	4,436 / 4,436
`squad`	15,820 / 19,940	15,820 / 15,820
`tldr`	2,826 / 19,811	2,826 / 2,826
`wp`	6,356 / 24,803	6,356 / 6,356
`xsum`	4,708 / 26,051	4,708 / 4,708
`yelp`	31,827 / 20,529	20,529 / 20,529

Table 5: The counts of documents for each domain for human and AI text before (left) and after (right) balancing. We balance the data such that each domain has an exactly 50:50 split.

Expert	**Ensemble weight**
xsum	0.181
roct	0.141
hswag	0.107
eli5	0.107
squad	0.094
cmv	0.093
wp	0.088
tldr	0.078
yelp	0.058
sci_gen	0.054

Table 6: Normalised logistic-regression coefficients showing each specialist's contribution to the "Weighted Vote" ensemble. Higher weights correspond to a stronger influence on the ensemble's output probability.

We observe that the per-domain balancing strategy consistently outperforms both dataset-wide and unbalanced training across in-distribution and out-of-distribution domains (see Table 4). As a result, we adopt per-domain balancing for all expert and baseline training throughout our experiments.

A.3 Hardware & Training Time

All models were trained using NVIDIA RTX A6000 GPUs with full-precision training. Training the baseline Qwen1.5–1.8B model used approximately 9.4 GPU hours, while the larger Qwen32B baseline required around 404.9 GPU hours. The QwenMoE model required approximately 402.3 GPU hours and training the domain experts, each based on Qwen1.5–1.8B, involved 10 domains and cumulatively consumed 235.1 GPU hours. The Joint Training models JT-Scratch and JT-Domain took an additional 861 GPU hours each. In total, model training across all baselines, experts, and ensembles required 2,773.7 GPU hours.

A.4 "Weighted Vote" Ensemble Weights

For the "Weighted Vote" comparison we train a Logistic Regression classifier using `scikit-learn` on the 10-dimensional vector of expert scores. We use the `lbfgs` solver, `max_iter=1000`, and `class_weight=balanced`. Prior to fitting, expert scores are standardised with `StandardScaler`. Table 6 reports the resulting normalised coefficients; they sum to 1 such that larger values indicate greater average influence on the final decision.

B Domains and Generators

In this section we briefly discuss the different domains and generators present in the MAGE (Li et al., 2024) and RAID (Dugan et al., 2024) benchmarks. We do this to help readers better understand the similarities and differences between the two domains and to give extra context on the transferability of our approach. We recommend reading the original papers for more detailed descriptions.

B.1 MAGE Domains

In Table 7 we report descriptions and sources for the 14 domains in MAGE. Of particular note is the DialogSum domain, which consists of natural conversations and is significantly different from other domains present in MAGE (see Appendix D.2). In general text was taken from publicly available sources such as Wikipedia, Reddit, Yelp, ArXiv, and News platforms (CNN, BBC, etc.).

B.2 MAGE Generators

In Table 8 we report the full list of 27 models used to generate the training split of the MAGE dataset.

Domain	Source Dataset	Description
CMV	r/ChangeMyView	Opinion statements from Reddit debates.
Yelp	Yelp Reviews	Restaurant and service reviews by users.
XSum	BBC Summaries	Short one-sentence news article summaries.
TLDR	TLDR_news	Concise news article summaries.
ELI5	Explain Like I'm 5	Long answers to simple crowd-asked questions.
WP	r/WritingPrompts	Creative stories from short prompts.
ROC	ROCStories	Commonsense, causal short stories.
HellaSwag	HellaSwag	Sentence completion for commonsense reasoning.
SQuAD	SQuAD	Wikipedia passages for QA context.
SciGen	SciXGen	Scientific article abstracts and prompts.
CNN	CNN/DailyMail	Articles paired with multi sentence summaries.
DialogSum	DialogSum	Realistic everyday conversations and summaries.
PubMedQA	PubMedQA	QA pairs from medical research literature.
IMDb	IMDb Reviews	Sentiment-rich movie reviews.

Table 7: Sources and descriptions for the 10 train and 4 held-out test domains included in the MAGE dataset.

Model	Identifier
OpenAI GPT (Brown et al., 2020)	`text-davinci-002`, `text-davinci-003`, `gpt-3.5-turbo`
LLaMA (Touvron et al., 2023)	`LLaMA-6B`, `LLaMA-13B` `LLaMA-30B`, `LLaMA-65B`
GLM (Zeng et al., 2022)	`GLM-130B`
FLAN-T5 (Chung et al., 2022)	`flan-t5-small`, `flan-t5-base`, `flan-t5-large`, `flan-t5-xl`, `flan-t5-xxl`
OPT (Zhang et al., 2022)	`opt-125M`, `opt-350M`, `opt-1.3B`, `opt-2.7B`, `opt-6.7B`, `opt-13B`, `opt-30B`, `opt-iml-1.3B`, `opt-iml-30B`
BigScience (Sanh et al., 2022)	`T0-3B`, `T0-11B`, `BLOOM-7B1`
EleutherAI (Black et al., 2022)	`GPT-J-6B`, `GPT-NeoX-20B`

Table 8: Full set of large language models used in the MAGE dataset. Each family is cited once and includes all released variants used in the dataset.

We see that the MAGE dataset is primarily made up of models that are pre-ChatGPT such as Llama-1 and FLAN-T5. We also see that the models used to generate are mainly on the smaller end (125M-7B) as compared to the average parameter count of more recent models. This makes it all the more surprising that DoGEN generalizes well to the RAID benchmark as it primarily consists of text generated by models released after ChatGPT.

In addition to these models, the authors of MAGE use GPT-4 (OpenAI, 2023) to generate examples for the held out domains (CNN, DialogSum, PubMed, IMDb) and paraphrase both the GPT-4 examples and the human-written examples with GPT3.5 (OpenAI, 2022). This makes the OOD domains significantly different from the main MAGE train distribution (see Appendix D).

Model	Identifier
GPT-2 (Radford et al., 2019)	`gpt2-xl`
MPT (+ *Chat*) (MosaicML, 2023)	`mpt-30b` `mpt-30b-chat`
Mistral (+ *Chat*) (Jiang et al., 2023)	`Mistral-7B-v0.1` `Mistral-7B-Instruct-v0.1`
LLaMA Chat (Touvron et al., 2023)	`Llama-2-70b-chat-hf`
Cohere (+ *Chat*) (Cohere, 2024)	`command (co.generate())` `command (co.chat())`
GPT-3 (Ouyang et al., 2022)	`text-davinci-002`
ChatGPT (OpenAI, 2022)	`gpt-3.5-turbo-0613`
GPT-4 (OpenAI, 2023)	`gpt-4-0613`

Table 9: The generative models used in the RAID dataset along with citations to the original papers.

B.3 RAID Domains

In Table 9 we list the domains present in the RAID dataset (Dugan et al., 2024). Many of the domains come from similar sources as MAGE. For example, the Wikipedia, Reddit, News, Abstracts, and Books domains all share common origins with domains present in MAGE. While this doesn't mean they are exactly the same distribution of text, it does means that we should expect ensembles trained on MAGE to perform relatively well on these splits.

The other domains that are dissimilar to MAGE are Reviews, Poetry, and Recipes. These are unusual sources for machine-generated text and, as expected, DoGEN does comparatively worse on these domains (see Table 2).

Domain	Source	Description
Abstracts	arxiv.org	ArXiv Abstracts
Recipes	allrecipes.com	Ingredients + Recipe
Books	wikipedia.org	Plot Summaries
Reddit	reddit.com	Reddit Posts
News	bbc.com/news	News Articles
Reviews	imbd.com	Movie Reviews
Poetry	poemhunter.com	Poems (Any Style)
Wiki	wikipedia.org	Article Introductions

Table 10: All domains in the RAID dataset alongside a description of where they are from. Clickable source links go directly to the source dataset from which the human samples were taken.

B.4 RAID Generators

In Table 9 we describe the models used to generate the RAID dataset. We see that most models are from mid-to-late 2023 and are likely quite different from the models present in MAGE.

Some similarities between the two sets of generators are present. For example, both datasets use text-davinci-002 and gpt-3.5-turbo. In addition, both datasets use a version of Llama for generation. However, RAID also includes models like Mistral, Cohere, and MPT, which are not included in either MAGE-train or test.

C Additional Expert Analysis

Tables 12 and 13 show that each specialist achieves its peak *AUROC* on domains it was fine-tuned for. We now examine whether the router exploits this specialisation.

The router allocates weight to the most suitable specialists. For every RAID document x we record

- the probability $p_i(x)$ assigned to expert E_i
- the stand-alone AUROC a_i that E_i attains on the full RAID corpus (Table 11, col. 2).

A positive association would indicate that the router systematically favours those specialists whose decision boundaries generalise best to the current input distribution. Figure 3 confirms this intuition (Pearson $\rho = 0.64,\ p < 0.03$): experts with larger AUROC attract more probability mass.

Take-away. Experts are neither intrinsically "good" nor "bad"—they are specialised to distinct domains. By first inferring the domain of each document, the router *learns to trust a specialist exactly when that specialist's domain matches the input* and to down-weight specialists that are out-of-distribution for the current example. This domain-aware routing allows the ensemble to generalise robustly to unseen genres without explicit supervision or hand-crafted rules.

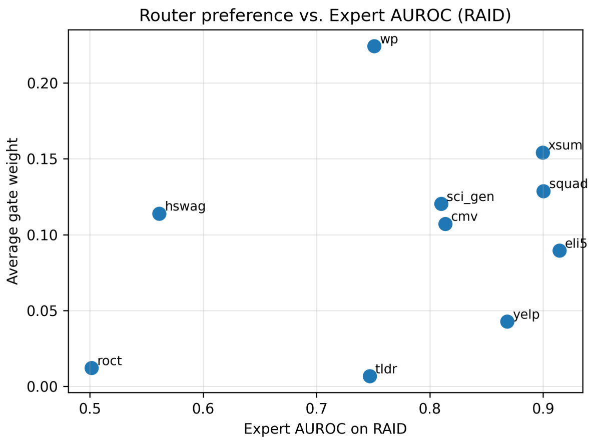

Figure 3: Average gate weight versus expert AUROC on RAID. Pearson $\rho = 0.64,\ p < 0.03$.

Expert	AUROC	Avg. $\bar{p}_i$	r_i
eli5	0.914	0.22	0.45
squad	0.846	0.19	0.40
cmv	0.814	0.17	0.35
sci_gen	0.810	0.16	0.32
yelp	0.804	0.15	0.30
xsum	0.760	0.12	0.25
tldr	0.742	0.10	0.20
wp	0.739	0.08	0.15
hswag	0.561	0.05	0.08
roct	0.501	0.03	0.05

Table 11: Standalone AUROC on RAID ("all" column), average gate weight $\bar{p}_i$, and per-expert Pearson correlation $r_i = \mathrm{corr}\big(p_i, \mathbf{1}\{\text{correct}\}\big)$. Higher-AUROC specialists receive larger weights and exhibit stronger positive correlations.

D Understanding Poor MAGE Out-of-Domain Performance

D.1 Paraphrased Data

The test domains in MAGE include text samples that differ meaningfully from the data used to train our expert models. While domain shift is expected in out-of-domain (OOD) evaluations, two of the three evaluation settings used in MAGE introduce particularly unusual cases.

Specifically, the `gpt4_para` setting includes outputs from GPT-4 that have been paraphrased by `gpt-3.5-turbo`. Likewise, the `human_para` setting consists of human-written text that has been paraphrased by `gpt-3.5-turbo`. These test

	MAGE (In-Domain)										**MAGE (Out-of-Domain)**				
Expert	cmv	eli5	tldr	xsum	wp	roct	hswag	yelp	squad	sci_gen	cnn	d_sum	imdb	pubmed	all
cmv	**0.998**	0.955	0.741	0.677	0.980	0.806	0.797	0.909	0.844	0.866	0.831	**0.970**	0.924	0.550	0.831
eli5	0.963	**0.989**	0.943	0.716	0.978	0.905	0.910	0.946	0.954	0.913	**0.897**	0.956	**0.977**	**0.753**	**0.897**
tldr	0.787	0.787	**0.995**	0.751	0.810	0.830	0.812	0.838	0.832	0.869	0.721	0.830	0.841	0.603	0.799
xsum	0.796	0.798	0.838	**0.995**	0.856	0.742	0.749	0.809	0.764	0.832	0.672	0.859	0.834	0.605	0.770
wp	0.875	0.842	0.772	0.688	**0.997**	0.765	0.772	0.810	0.800	0.850	0.732	0.840	0.818	0.602	0.781
roct	0.444	0.579	0.798	0.427	0.507	**0.989**	0.846	0.577	0.642	0.688	0.654	0.488	0.487	0.506	0.654
hswag	0.457	0.614	0.678	0.400	0.430	0.564	**0.992**	0.593	0.709	0.697	0.608	0.962	0.498	0.488	0.608
yelp	0.842	0.871	0.867	0.744	0.890	0.813	0.799	**0.996**	0.851	0.904	0.736	0.913	0.866	0.671	0.814
squad	0.946	0.932	0.899	0.781	0.948	0.776	0.786	0.933	**0.982**	0.899	0.840	0.935	0.917	0.717	0.857
sci_gen	0.905	0.790	0.863	0.746	0.881	0.749	0.833	0.846	0.907	**0.971**	0.803	0.962	0.846	0.636	0.803

Table 12: Expert-Level AUROC Results on MAGE. Columns grouped into In-Domain and Out-of-Domain (OOD) Best performance in each column is bolded. d_sum is shorthand for dialog_sum

	RAID (Out-of-Domain) – AUROC								
Expert	abstracts	books	news	poetry	recipes	reddit	reviews	wiki	**all**
cmv_expert	**0.970**	0.924	0.550	0.896	0.796	0.931	0.946	0.580	0.814
eli5_expert	0.956	**0.977**	**0.753**	**0.943**	**0.939**	**0.948**	**0.973**	**0.861**	**0.914**
tldr_expert	0.804	0.791	0.644	0.702	0.739	0.760	0.778	0.719	0.742
xsum_expert	0.812	0.801	0.676	0.747	0.765	0.760	0.784	0.729	0.760
wp_expert	0.834	0.782	0.639	0.697	0.743	0.755	0.776	0.690	0.739
roct_expert	0.488	0.487	0.506	0.535	0.489	0.493	0.533	0.480	0.501
hswag_expert	0.962	0.498	0.488	0.533	0.496	0.492	0.524	0.504	0.561
yelp_expert	0.876	0.864	0.697	0.782	0.805	0.815	0.838	0.753	0.804
squad_expert	0.905	0.854	0.730	0.791	0.864	0.870	0.905	0.852	0.846
sci_gen_expert	0.962	0.846	0.636	0.610	0.895	0.776	0.898	0.828	0.810

Table 13: Expert-Level AUROC Results on RAID (Out-of-Domain). Best performance in each column is bolded.

cases do not cleanly align with either "machine-generated" or "human-written" categories and introduce ambiguity into the labeling scheme.

As shown in Table 14, our model actually performs quite well at detecting `human_para` text as human-written. This result supports our position that such examples—although technically modified by a machine—are stylistically closer to human writing and their inclusion in the "machine" class may be misleading. We argue that the use of paraphrased human text as a negative class label introduces noise into the evaluation and may inflate or deflate detector performance in ways that are difficult to interpret.

Despite this ambiguity, DoGEN performs competitively on `gpt4` examples—an especially relevant case, as GPT-4 was not seen by any expert during training. This suggests that DoGEN is capable of generalizing to unseen generators and adapting to challenging OOD distributions, performing comparably or better than other baselines.

D.2 Challenges with the `dialogsum` Domain

One domain where our models notably underperform is `dialogsum`. This domain differs significantly in structure from our in-domain training sets. All of our in-domain examples (e.g., `cmv`, `eli5`, `xsum`) are prose-style summaries or longform posts, typically formatted as block paragraphs of explanatory or narrative text. In contrast, `dialogsum` consists of multi-speaker conversational transcripts, with speaker tags, turn-taking, short utterances, and question–answer structure.

Our experts were never trained on dialogue-style input. Consequently, the gating classifier has difficulty assigning dialogsum inputs to the correct expert, and the experts themselves struggle to map multi-turn dialogue into effective summaries. This mismatch leads to weaker performance on this domain. An example of such dialog-style input is shown in Table 16.

D.3 Adapting with DoGEN

A key strength of the DoGEN architecture is its modularity. Because each expert is independently

	CNN				**DialogSum**				**IMDb**				**PubMedQA**			
	gpt4	gpt4-p	hum-p	all	gpt4	gpt4-p	hum-p	all	gpt4	gpt4-p	hum-p	all	gpt4	gpt4-p	hum-p	all
Qwen1.8B	71.6	66.7	54.0	71.6	61.8	31.9	43.4	39.1	91.5	95.9	79.3	91.6	82.0	88.1	58.1	82.0
Qwen32B	**98.6**	**95.9**	**86.1**	**93.5**	**94.8**	**91.7**	**85.0**	**90.5**	97.7	96.7	87.4	93.9	96.1	**93.4**	**83.5**	**91.0**
QwenMoE	91.2	89.6	78.8	86.5	91.0	80.5	53.8	75.1	**98.9**	**97.7**	**91.6**	**96.1**	98.6	89.5	74.9	87.7
Equal Vt.	87.3	37.9	36.2	53.8	91.6	37.1	25.4	38.0	95.8	90.6	71.6	86.0	**99.2**	90.2	63.3	84.2
Weight Vt.	91.5	64.5	40.3	65.4	82.7	38.8	26.8	39.4	96.1	88.5	71.3	82.3	99.0	84.5	61.8	80.1
JT-Domain	86.2	53.0	45.1	61.6	86.6	39.8	51.1	62.5	98.3	96.7	79.9	91.6	94.7	79.9	57.5	77.4
JT-Scratch	90.7	70.6	68.5	76.6	85.4	62.2	39.3	62.3	95.3	90.8	78.5	81.2	97.7	83.8	62.4	80.3
DoGEN	97.1	85.5	78.3	78.9	73.3	51.5	32.1	45.6	95.3	93.2	78.3	84.9	97.2	84.2	78.9	79.1

Table 14: AUROC (%) scores for MAGE Out-of-Domain. "gpt4-p" and "hum-p" refer to gpt4 and human texts paraphrased by `gpt-3.5-turbo` (see Section B.2). We see that while DoGEN does poorly on aggregate metrics, on gpt4 generations it does well in every domain except dialog_sum (see Section D.2).

RAID – TPR@FPR=5%									
Model	abstracts	books	news	poetry	recipes	reddit	reviews	wiki	all
Qwen1.8B	85.01	84.44	84.19	82.59	74.22	81.79	85.26	83.56	82.63
Qwen32B	**91.85**	86.66	91.62	82.41	**86.97**	73.34	88.38	75.10	84.54
QwenMoE	72.93	90.46	84.25	79.03	82.84	85.65	**93.44**	87.10	84.46
Equal Vote	87.68	90.68	87.79	68.84	72.41	**89.07**	90.38	82.62	83.68
Weighted Vote	86.24	88.24	86.12	70.89	75.13	79.57	80.20	83.11	83.75
JT-Domain	90.63	**91.65**	91.56	80.94	21.41	86.69	88.82	84.34	79.51
JT-Scratch	89.63	89.10	**95.28**	**92.40**	39.70	85.60	81.96	87.01	82.58
DoGEN (ours)	89.28	91.13	91.76	75.16	79.81	82.82	89.76	**87.16**	**85.86**

Table 15: True Positive Rate at 5% False Positive Rate (TPR@FPR=5%) across RAID domains. Bold values indicate the best score in each column.

Example from the `dialogsum` domain:
"Person1: Hello. Person2: Hello. May I speak to Mark, please? Person1: I apologize, but Mark is not currently available. Would you like me to take a message or assist you with something else?"

Table 16: Quoted example from the `dialogsum` domain.

trainable, we can train a new expert specifically on `dialogsum` examples and retrain the domain classifier to incorporate it. This offers a principled and efficient way to extend the system to structurally novel inputs without the need to retrain all experts or overhaul the entire ensemble.

By addressing domain-specific structural mismatches and class imbalance, we ensure that our evaluation of generalization is both fair and adaptable to new domains.

E Additional Results

RAID Results TPR@FPR=5% For our evaluation on RAID we also report **True Positive Rate at a 5% False Positive Rate** (TPR@FPR= 5%) in Table 15. This quantifies how effectively a detector identifies machine-generated text while limiting false positives on human-written content to 5%. We compute this metric using threshold values obtained via the RAID benchmark API across all evaluation domains. We do this to match the more standard evaluation metric that is calculated for submission to the RAID benchmark allowing us to be comparable to other established detection methods.

F AI Assistance

During the writing of this paper AI assistants were used to help enhance clarity of writing, format tables and figures, as well as draft language for the formalisms in the methods section. The assistants used were ChatGPT and OpenAI o3.