All in the Head?: A Controlled Study of Component Contributions in Few-Shot NLP

Anonymous ACL submission

Abstract

Few-shot text classification is often studied through model scaling or full fine-tuning, but less is known about how classification head design influences performance when representations are held fixed. This work examines that question under a controlled frozen-encoder setting, where a compact LSTM-based head is trained on top of contextual embeddings while all encoder parameters remain unchanged. Rather than proposing a new architecture, we use this setting as a controlled evaluation of how head aggregation, attention, and lexical perturbation affect low-shot classification when token representations are fixed. Across datasets, removing attention reduces accuracy by 5.1 points on average, removing augmentation reduces accuracy by 3.0 points, and removing both reduces accuracy by 8.2 points. We report consistent improvements over simpler head configurations, including a direct mean-pooling MLP baseline, and competitive performance relative to compact transformer-based alternatives under identical conditions, while maintaining a low optimization footprint. These results provide evidence that head architecture and training choices remain consequential even with fixed contextual encoders, and highlight a simple controlled framework for studying inductive biases in low-shot classification systems.

1 Introduction

Few-shot text classification sits at the intersection of two practical constraints, limited labeled data and limited optimization and deployment budgets. Large language models enable few-shot behavior through prompting (Brown et al., 2020), and parameter-efficient adaptation methods update only a small number of parameters (Houlsby et al., 2019; Lester et al., 2021; Li and Liang, 2021). In many deployments, these approaches compete with a simpler alternative: using a strong frozen encoder to produce contextual token representations and training a compact task head on top. This setting reduces training memory and provides a clean control for studying what head architectures contribute when representations are held fixed.

We revisit this controlled setting with a compact LSTM-based classification head. Recurrent architectures introduce a sequential inductive bias that can help aggregate token-level evidence, while lightweight attention provides a mechanism for emphasizing salient tokens (Bahdanau et al., 2015). Because modern contextual encoders such as BERT already encode substantial syntactic and semantic structure (Devlin et al., 2019), this motivates a concrete question: under a fixed contextual encoder, which head components reliably improve low-shot classification, and how large are those effects? By freezing the encoder and varying only the trained head and low-shot perturbation pipeline, we isolate architectural effects that are often hidden inside full fine-tuning or prompting comparisons.

To study this, we combine a compact LSTM over frozen BERT token embeddings with a lightweight additive attention layer and targeted synonym-based augmentation (Wei and Zou, 2019; Miller, 1994). The encoder remains frozen and all optimization occurs in the head. We evaluate across SST-2 and SST-5 (Socher et al., 2013), RAFT (Alex et al., 2021), and AGNews,[1] using a consistent few-shot protocol with 21–25 labeled examples per dataset and comparisons to compact transformer-family baselines and a transformer-head replacement under the same frozen-encoder protocol.

Our results show that a carefully structured LSTM head remains competitive under tight data constraints while keeping the optimization footprint small, with consistent gains from both attention and targeted augmentation.

2 Our Approach

We study a controlled few-shot setting where a strong contextual encoder is frozen and only a compact classification head is trained. As shown in Fig. 1, we expand each labeled example with targeted synonym-based augmentation, embed each variant with BERT (Devlin et al., 2019), and train an LSTM-based head with additive attention (Bahdanau et al., 2015). Gradients are never propagated through the encoder (the embedding forward pass is wrapped in `torch.no_grad()`), so all optimization occurs in the head. This design isolates head behavior under fixed representations and lets us report efficiency in terms of trainable parameters and head compute, while separately noting the full encoder footprint.

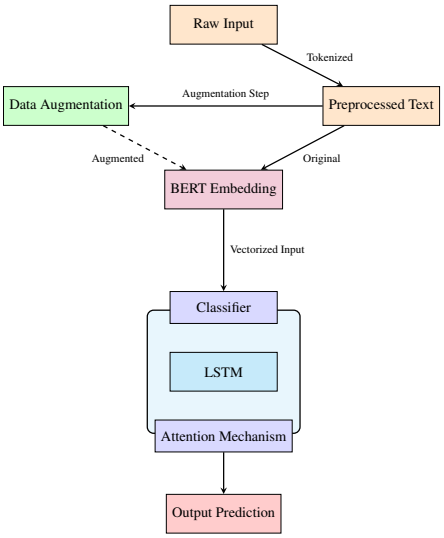

Figure 1: Flow diagram illustrating the process from raw input to output prediction, incorporating preprocessing, data augmentation, BERT embedding, and an LSTM module consisting of a classifier and attention mechanism.

2.1 Targeted Synonym Augmentation

To increase lexical diversity without large semantic drift, we use synonym replacement from WordNet (Miller, 1994). For a tokenized sentence $\langle w_1, w_2, \ldots, w_n \rangle$, we replace a small subset of content words with WordNet synonyms:

$$s' = \{w_1, \ldots, \text{Synonym}(w_k), \ldots, w_n\}, \quad w_k \in s.$$

We restrict candidates to nouns, verbs, adjectives, and adverbs and sample replacements in proportion to part-of-speech frequency in the sentence. This yields multiple variants per labeled example and expands the effective training set used to fit the head.

2.2 LSTM Head with Additive Attention

Given BERT token embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ for an input sentence, the head applies an LSTM to produce hidden states $\{\mathbf{h}_1, \ldots, \mathbf{h}_n\}$:

$$\mathbf{h}_i = \text{LSTM}(\mathbf{e}_i, \mathbf{h}_{i-1}), \quad \mathbf{h}_i \in \mathbb{R}^h.$$

We compute additive attention scores and weights,

$$e_i = \mathbf{v}^\top \tanh(\mathbf{W}\mathbf{h}_i), \qquad \alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)},$$

and form a single context vector,

$$\mathbf{c} = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i.$$

The classifier predicts class probabilities from $\mathbf{c}$:

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_o \mathbf{c} + \mathbf{b}),$$

as illustrated in Fig. 2.

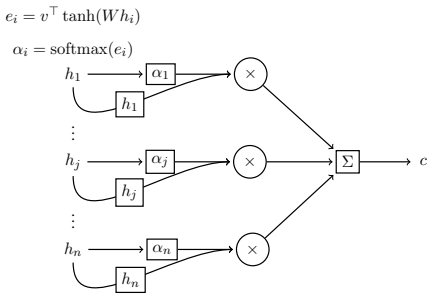

Figure 2: Diagram of the attention mechanism in the LSTM model. Attention weights α_i are computed using $e_i = v^\top \tanh(Wh_i)$, followed by a softmax. Here W and v are learnable parameters and h_i is the LSTM hidden state.

2.3 Optimization

We train the head with cross-entropy loss and label smoothing:

$$\mathcal{L} = -\sum_{i=1}^{k} y_i \log(\hat{y}_i).$$

Optimization uses Adam (Kingma and Ba, 2014) with learning rate 6.69×10^{-4}, cosine annealing, and gradient clipping. Label smoothing is set to 0.1 and held fixed across datasets. All reported results are averaged over 5 random seeds.

3 Experiments

Datasets and few-shot splits. We evaluate on SST-2 and SST-5 for sentiment (Socher et al., 2013), RAFT for diverse real-world classification tasks (Alex et al., 2021), and AGNews for topic classification. We use 25-shot training for SST-2, RAFT, and AGNews, and 21-shot training for SST-5. For each dataset, we construct few-shot training sets by sampling labeled examples per class under a fixed protocol and report performance averaged over 5 random seeds.

Head variants and controlled toggles. We study head design under a fixed representation function. For all experiments, we use `bert-base-uncased` as a frozen contextual encoder and never backpropagate through it. Given an input sequence, we extract token-level embeddings and train only a compact classification head. Our primary head is a compact LSTM over frozen token embeddings, followed by a single-head additive attention mechanism that produces a context vector, and a linear classifier. We evaluate four configurations under identical encoder, optimizer, and data conditions: (1) full head (LSTM + attention + synonym augmentation). (2) no attention (remove attention and classify using the LSTM output aggregation used in our implementation). (3) no augmentation (disable synonym-based augmentation while keeping the full LSTM + attention head). (4) neither (disable both attention and augmentation). Synonym augmentation uses WordNet (Miller, 1994). We additionally evaluate a mean-pooling MLP baseline that averages frozen BERT token embeddings and feeds the pooled representation to an MLP classifier.

Significance testing. For the main component comparisons, we compute approximate Welch tests from the reported mean and standard deviation over five seeds, comparing the full head against each reduced variant for each dataset. We report these tests as secondary evidence alongside the effect sizes, since the main claim concerns the direction and magnitude of component contributions across datasets.

Baselines and prompting comparisons. Expanded baseline comparisons, including a mean-pooling MLP baseline, compact transformer-family models, and a transformer-head replacement under the same frozen-encoder protocol, are reported in Appendix A. Prompting-based comparisons (GPT-3 and GPT-4o), along with prompt templates and evaluation details, are reported in Appendix B.

4 Results

Our results are organized around a single controlled question: under a frozen contextual encoder, which inductive biases in the classification head and training pipeline yield the most reliable gains in few-shot text classification?

Table 1 summarizes our primary controlled comparison across all datasets. The full head (LSTM + attention + augmentation) performs best overall, and removing either attention or augmentation consistently reduces accuracy. Removing both components produces the largest drop, indicating that the gains are not attributable to a single factor. We quantify these effects more directly in Table 2.

Model Variant	SST-2 (25-shot)	SST-5 (21-shot)
Full head (ours)	**75.1%** ($\pm$1.7)	**83.4%** ($\pm$2.3)
No attention	71.1% ($\pm$2.0)	78.9% ($\pm$2.5)
No augmentation[2]	72.0% ($\pm$2.2)	81.2% ($\pm$2.1)
Neither	66.4% ($\pm$2.5)	74.0% ($\pm$2.7)
	RAFT (25-shot)	**AGNews (25-shot)**
Full head (ours)	**81.9%** ($\pm$2.2)	**87.6%** ($\pm$2.0)
No attention	77.3% ($\pm$2.4)	80.2% ($\pm$2.1)
No augmentation	80.1% ($\pm$2.0)	82.6% ($\pm$2.3)
Neither	76.1% ($\pm$2.4)	78.7% ($\pm$2.5)

Table 1: Component impact under a frozen encoder (accuracy, mean $\pm$ std over 5 seeds). The full head uses an LSTM with additive attention and synonym augmentation. Variants remove attention, augmentation, or both, while keeping the frozen encoder and the rest of the protocol fixed.

Aggregating across datasets gives a compact estimate of each component's contribution. Removing attention lowers accuracy by 5.1 points on average, removing augmentation lowers accuracy by 3.0 points, and removing both lowers accuracy by 8.2 points. The full head also has the lowest average standard deviation across seeds (2.05), while the variant without either component has the highest (2.53), suggesting that the complete head improves both mean performance and low-shot stability.

[1] https://huggingface.co/datasets/sh0416/ag_news

[2] Augmentation denotes synonym replacement with WordNet under the constraints described in Section 2.

Variant	Mean accuracy drop	Mean seed std.
No attention	−5.1	2.25
No augmentation	−3.0	2.15
Neither	−8.2	2.53
Full head	–	2.05

Table 2: Aggregate component effects across SST-2, SST-5, RAFT, and AGNews. Accuracy drops are computed relative to the full LSTM head.

Because the encoder is frozen, the primary optimization footprint is the trained head. The LSTM head contains 3.09M trainable parameters, uses 396.83 MFLOPs, and takes 55.90 ms per instance on CPU. Including `bert-base-uncased`, the full parameter count is approximately 113M. A broader efficiency comparison is provided in Appendix C.

Approximate Welch tests from reported seed means and standard deviations support the main effects: attention removal and removal of both components are significant across all datasets, while augmentation removal is significant on SST-2 and AGNews. The augmentation effect is smaller and less uniform than the attention effect, suggesting that lexical perturbation is useful but more task-dependent. Full results are in Appendix F.

The mean-pooling MLP baseline in Appendix E checks whether the gains come only from frozen BERT representations. The full LSTM-based head outperforms direct pooling on SST-2, SST-5, and AGNews by 8.9, 57.0, and 11.5 points, respectively, suggesting that sequence aggregation and attention contribute beyond pooled contextual embeddings, although the especially large SST-5 gap should be interpreted in light of the weak direct-pooling baseline. For RAFT, the MLP result uses an internal split of the labeled training set because official labels are unavailable for the public test split.

5 Discussion

Our results show that frozen contextual representations do not make the classification head irrelevant. Choices in the head and low-shot pipeline still drive meaningful performance differences: removing attention lowers accuracy by 5.1 points on average, removing augmentation lowers accuracy by 3.0 points, and removing both lowers accuracy by 8.2 points. Approximate Welch tests in Appendix F provide secondary support for these effects.

A compact LSTM head concentrates optimization where few-shot training is most sensitive. In our setting, where the encoder is frozen and only a small number of parameters are updated, the head is a clean interface from token-level representations to a sentence-level decision, making it straightforward to compare architectural choices under identical upstream features. The mean-pooling MLP baseline further separates the contribution of the frozen encoder from the contribution of sequence aggregation in the trained head.

Attention also appears to interact favorably with synonym augmentation. Even with constrained substitutions, synonym replacement can introduce occasional lexical noise. A lightweight additive attention layer gives the head a mechanism to emphasize predictive tokens and down-weight less useful perturbations, aligning with the consistent drops when attention is removed. The mean-pooling MLP and transformer-head replacement results in Appendix E further suggest that neither direct pooling nor swapping in a different lightweight sequence model is guaranteed to match the LSTM head.

Limitations

Augmentation relies on English WordNet and we evaluate only English, single-label tasks, so the exact pipeline does not necessarily transfer to low-resource languages or multi-label settings without modification. The controlled head comparisons remain valid under these constraints, but broader applicability requires empirical validation. A natural extension keeps the study design intact while swapping components: replace WordNet with language-specific or embedding-based augmentation, and replace BERT with multilingual encoders (e.g., mBERT or XLM-R).

Conclusion

Our study examines head design for few-shot text classification under a fixed representation function. We do so by freezing BERT, training a compact LSTM-based head, and isolating the effects of additive attention and targeted synonym augmentation under the same low-shot protocol. Across multiple datasets, the LSTM-based head benefits consistently from both components, with the largest performance drop occurring when attention and augmentation are removed together. The mean-pooling MLP and transformer-head comparisons further suggest that these gains are tied to how token-level evidence is aggregated, beyond the strength of the encoder alone. Overall, these results support frozen-encoder head studies as a useful way to isolate architectural contributions in few-shot NLP.

References

Neel Alex and 1 others. 2021. Raft: A real-world few-shot text classification benchmark. In *Advances in Neural Information Processing Systems*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Tom Brown and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Jacob Devlin and 1 others. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, and Fang Wang. 2020. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:2006.08239*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

George A. Miller. 1994. Wordnet: A lexical database for english. In *Proceedings of the International Conference on Human Language Technology*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Philipp Schmid. 2022. Setfit ag news endpoint. https://huggingface.co/philschmid/setfit-ag-news-endpoint. Accessed: 2025-05-04.

Richard Socher and 1 others. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. In *Proceedings of the Second Workshop on Efficient Natural Language and Speech Processing (ENLSP) at NeurIPS 2022*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of EMNLP*.

A Appendix: Baselines

We report additional baseline results to contextualize the frozen-encoder head study in the main paper. All baselines follow the same few-shot split construction and are averaged over 5 random seeds unless otherwise noted. For encoder-based baselines, we use the corresponding model as specified by its standard checkpoint. For the mean-pooling MLP and transformer-head replacements, we keep the frozen encoder fixed and swap only the trained head while preserving the rest of the protocol.

Full per-dataset metrics and the complete expanded table are provided in Appendix E.

B Appendix: Prompting details

To situate the controlled component study in a broader context, Table 3 reports accuracy comparisons across datasets. The full head achieves strong performance under the frozen-encoder protocol. Expanded baseline comparisons, including compact transformer-family models and a transformer-head replacement under identical frozen-encoder conditions, are reported in Appendix A.

Model	SST-2 (25-shot)	SST-5 (21-shot)	RAFT (25-shot)	AGNews (25-shot)
Ours (full head)	**75.1%** (±1.7)	**83.4%** (±2.3)	**81.9%** (±2.2)	**87.6%** (±2.0)
GPT-3[3]	–	66.0%	68.6%	–
GPT-4o[4]	89.9% (±1.9)	8.5% (±3.2)	81.4% (±2.0)	44.8% (±2.4)
SetFit[5]	–	43.6%	71.3%	87.6%

Table 3: Accuracy comparison across datasets (mean ± std over 5 seeds).

[3]Metric taken from an existing benchmark (Alex et al., 2021). Since OpenAI no longer publicly allows access to GPT-3, we were unable to conduct additional tests.

[4]Prompting setup is reported in Appendix B.

[5]SetFit introduced in (Tunstall et al., 2022). AGNews result taken from (Schmid, 2022).

We include prompting-based comparisons (GPT-3 and GPT-4o) for broader context, but these results do not follow the frozen-encoder training protocol and should not be interpreted as controlled head comparisons. For GPT-3, we cite the RAFT benchmark numbers reported in Alex et al. (2021), since GPT-3 is no longer publicly accessible for reruns. For GPT-4o, we evaluated each example independently using a fixed prompt template per dataset (listed in Appendix D), and we parsed the output by mapping the model's predicted label string to the task label set. We ran multiple trials and verified that label mappings were consistent with the dataset label definitions. We also validated that the evaluation script correctly aligned predictions and labels and that the same parsing code produced expected results on simple inputs.

We observed an unusually low GPT-4o accuracy on SST-5 and treat it as an artifact of this prompting setup rather than a core claim.

C Appendix: Efficiency comparison

Model	Params	FLOPs	Latency (CPU)	Latency (GPU)
GPT-4o	> 1T[6]	Unknown	N/A	N/A
DistilBERT[7]	66M	∼1.2 GFLOPs	∼90 ms	∼30 ms[8]
MobileBERT (MB)[9]	25M	∼1.3 GFLOPs	∼80 ms	∼24 ms
TinyBERT[10]	14M	∼1.3 GFLOPs	∼70 ms	∼22 ms
Ours (head)[11]	3.09M	396.83 MFLOPs	55.90 ms	9.62 ms

Table 4: Model efficiency comparison.

D Appendix: Prompt Templates and Reporting Details

For transparency and reproducibility, we include the final prompt templates used for GPT-4o and any other prompted LLM baselines in this appendix. The results in Table 3 use the fixed prompt templates below.

D.1 Prompt for AGNews

You are a news article classifier. Your task is to classify news articles into one of four classes:

0: World
1: Sports
2: Business
3: Science/Technology

D.2 Prompt for RAFT's ADE Corpus

You are a medical text classifier. Your task is to classify sentences into one of two classes:

1: Adverse Event
2: No Adverse Event

D.3 Prompt for SST-2

You are a sentiment classifier. Your task is to classify sentences into one of two classes:

0: Negative sentiment
1: Positive sentiment

D.4 Prompt for SST-5

You are a sentiment classifier. Your task is to classify sentences into one of five classes:

0: Very Negative sentiment
1: Negative sentiment
2: Neutral sentiment
3: Positive sentiment
4: Very Positive sentiment

E Appendix: Expanded baselines and additional metrics

The mean-pooling MLP baseline averages frozen BERT token embeddings and feeds the pooled representation to an MLP classifier. For RAFT, the MLP result is reported using an internal split of the 50 labeled training examples because official labels are unavailable for the public test split, so this row should be interpreted as an internal-split diagnostic rather than a direct comparison to the main RAFT result.

Model	Dataset	Accuracy	F1	Precision / Recall
DistilBERT	SST-2	50.1%	0.501	0.501 / 0.501
DistilBERT	SST-5	13.8%	0.236	0.996 / 0.138
DistilBERT	AGNews	29.6%	0.183	0.293 / 0.296
DistilBERT	RAFT	60.0%	0.688	0.917 / 0.550
TinyBERT	SST-2	53.2%	0.448	0.604 / 0.532
TinyBERT	SST-5	48.7%	0.650	0.994 / 0.487
TinyBERT	AGNews	35.4%	0.289	0.366 / 0.354
TinyBERT	RAFT	68.0%	0.790	0.833 / 0.750
MobileBERT	SST-2	67.2%	0.659	0.711 / 0.672
MobileBERT	SST-5	70.8%	0.825	0.993 / 0.708
MobileBERT	AGNews	80.8%	0.818	0.798 / 0.818
MobileBERT	RAFT	68.0%	0.810	0.773 / 0.850

Table 5: Expanded encoder-family baselines under the same few-shot protocol.

[6] Estimate, not publicly released.
[7] DistilBERT (Sanh et al., 2019). FLOPs and latency based on the official paper.
[8] Based on HuggingFace benchmarks.
[9] MobileBERT (Sun et al., 2020). FLOPs and latency based on community benchmarks and the official paper.
[10] TinyBERT (Jiao et al., 2020). Latency from HuggingFace ONNX and PyTorch benchmarks.
[11] Head-only metrics. Full parameter count including frozen encoder is approximately 113M for `bert-base-uncased` plus head.

Model	Dataset	Accuracy	F1	Precision / Recall
Mean-pooling MLP	SST-2	66.2%	0.650	0.680 / 0.661
Mean-pooling MLP	SST-5	26.4%	0.259	0.280 / 0.290
Mean-pooling MLP[12]	RAFT	59.2%	0.487	0.578 / 0.733
Mean-pooling MLP	AGNews	76.1%	0.761	0.785 / 0.761
Transformer head[13]	SST-2	53.4%	0.520	0.535 / 0.534
Transformer head	SST-5	61.8%	0.760	0.991 / 0.618
Transformer head	RAFT	67.6%	0.800	0.800 / 0.800
Transformer head	AGNews	54.7%	0.543	0.547 / 0.547
Ours (head)[14]	SST-2	75.1%	–	–
Ours (head)	SST-5	83.4%	–	–
Ours (head)	RAFT	81.9%	–	–
Ours (head)	AGNews	87.6%	–	–

Table 6: Expanded head-family baselines and additional metrics under the same few-shot protocol.

F Appendix: Significance testing

We report approximate Welch tests computed from the reported mean and standard deviation over five seeds. These tests should be interpreted as secondary evidence alongside the effect sizes in the main text.

Comparison	SST-2	SST-5	RAFT	AGNews
Full vs. No attention	0.0096 (4.0)	0.0182 (4.5)	0.0135 (4.6)	0.0005 (7.4)
Full vs. No augmentation	0.0391 (3.1)	0.1532 (2.2)	0.2131 (1.8)	0.0065 (5.0)
Full vs. Neither	0.0003 (8.7)	0.0004 (9.4)	0.0041 (5.8)	0.0003 (8.9)

Table 7: Approximate Welch tests computed from the reported mean and standard deviation over five seeds. Entries show p-values, with mean accuracy differences in points shown in parentheses.

[12] RAFT MLP result uses an internal split of the 50 labeled RAFT training examples, since public test labels are not directly available in the dataset.

[13] Transformer head replacement of our LSTM head, keeping the frozen encoder and the rest of the protocol identical.

[14] Accuracy matches Table 3 in the main paper.