

Encoding Task Structure via Attention Biases and Adaptive Computation

Stanford CS224N Default Project

Dario Soatto

Department of Computer Science
Stanford University
soatto@stanford.edu

Abstract

Transformer pre-training and fine-tuning treat all positions and token interactions uniformly even when the downstream task has known structure. However, in situations with limited compute or memory resources, existing generalized fine tuning processes may not be able to fully exploit this known task structure. We investigate whether explicitly encoding known task structure into the attention mechanism of GPT-2 can improve fine-tuning performance. We propose two extensions: (1) task-specific attention bias matrices that encourage structurally relevant attention patterns, and (2) iterative computation that re-runs final transformer layers at critical positions. We evaluate on paraphrase detection (Quora Question Pairs) and sonnet generation (Shakespeare). Attention biases yield consistent improvements when model capacity is constrained (up to +0.9% accuracy on paraphrase detection and +1.64 ChrF on sonnet generation) while a random-mask ablation confirms that structured geometry is essential. Iterative computation does not improve performance, which we attribute to a distribution mismatch when re-running pre-trained layers outside their expected input distribution. Our results show that the value of injected inductive biases depends critically on the fine-tuning regime: they help when the model cannot freely adapt its attention patterns, but are redundant under full fine-tuning. This makes attention biases a practical, near-zero-cost tool for improving performance under resource constraints, particularly where full fine-tuning is infeasible.

1 Key Information to include

- Mentor: Alisa Levin
- External Collaborators (if you have any) or sharing project: N/A
- Late Days (if you are using any): 3 (solo project, not pooled)

2 Introduction

Standard transformer models treat all token positions uniformly: each position attends to previous positions based solely on learned query-key compatibility, and every token receives the same amount of computation (i.e., a fixed number of layers). However, many NLP tasks have predictable structures that are explicitly known *a priori* and inform where the model should attend and how much computation different positions require. Although fine-tuning may help the model learn the structure of a task implicitly, in settings with limited data or constrained fine-tuning capacity, learning the proper structure may be infeasible, thereby leaving potential gains from exploiting the known structure on the table.

We investigate whether explicitly encoding known task structure into the attention mechanism of GPT-2 [1] can improve fine-tuning performance. We study two structurally distinct tasks: *cloze-style paraphrase detection*, where determining whether two sentences are paraphrases requires explicit

cross-sentence comparison, and *sonnet generation*, where the model must condition strongly on a given prompt to produce coherent continuation lines.

We propose two extensions. First, we add task-specific bias matrices to the attention scores, drawing on ALiBi [2] and Attention Bias Scaffolding [3], to encourage structurally relevant attention patterns. Second, inspired by Adaptive Computation Time [4] and PonderNet [5], we allocate additional computation to the classification decision of the paraphrasing task by re-running the final transformer layers multiple times; we explore variants that refine only the classification token and variants that update all positions.

Our experiments reveal a consistent pattern: attention biases improve performance when model capacity is constrained (e.g., frozen backbone), with learnable per-head bias variants yielding up to +0.9% accuracy on paraphrase detection and fixed biases improving sonnet ChrF (character n -gram F-score) by up to +1.64 under partial fine-tuning. A random-mask ablation confirms that these gains stem from the encoded structure rather than increased parameter count. Under full fine-tuning, the model learns effective attention patterns on its own and the biases provide little additional benefit. Iterative computation does not improve performance in any setting, which we attribute to a distribution mismatch: pre-trained layers expect input from the preceding layer, but repeating certain transformer layers breaks this pattern. Prior literature on adaptive computation canonically involves pre-training the transformer with the adaptive computation [6]; we find that fine-tuning alone is not enough to overcome the distribution mismatch.

The remainder of this paper is organized as follows. Section 3 discusses related work on attention biases and adaptive computation. Section 4 details our approach, including both extensions and their variants. Section 5 presents our experimental setup and results. Section 6 provides analysis of when and why each extension succeeds or fails, and Section 7 concludes. An appendix provides further analysis of attention patterns and qualitative comparison of generated sonnets.

3 Related Work

Attention biases as inductive bias. The transformer’s self-attention mechanism is permutation-invariant, so encoding positional or structural information requires an explicit mechanism. Early approaches addressed this through additive position representations: sinusoidal embeddings [7], relative position encodings [8], rotary embeddings [9], and learned pairwise biases in T5 [10]. These methods share the goal of injecting positional awareness but all encode generic and generalizable notions of position (e.g., absolute index or relative distance) rather than task-specific structure.

ALiBi [2] demonstrated that a fixed, non-learned linear penalty added directly to attention scores rather than to input embeddings is sufficient for strong length extrapolation, establishing attention-score biases as a lightweight and effective intervention point. Duan et al. [3] took this idea further by moving from generic positional biases to task-specific structural ones: they diagnosed vanilla transformers’ failure on arithmetic as a consequence of inadequate inductive biasing and showed that hand-crafted attention masks enforcing which digit positions attend to one another result in complete length generalization on addition and parity tasks. However, their work focused on synthetic algorithmic tasks with encoder-decoders trained from scratch, leaving open the question of whether similar structural biases can improve fine-tuning of pre-trained language models on natural language tasks. Our work bridges this gap: we adopt ALiBi’s mechanism of adding bias matrices to attention scores but replace the distance-based penalty with task-specific structural priors designed for paraphrase detection and sonnet generation, applied during fine-tuning of a pre-trained GPT-2 model. To our knowledge, no prior work has applied task-specific attention biases during fine-tuning of a pre-trained decoder-only language model on natural language tasks.

Adaptive and iterative computation. Standard transformers apply a fixed computational budget to every token, regardless of the "difficulty" of correctly predicting that token. Adaptive Computation Time (ACT) [4] introduced the idea of dynamically varying computation depth via a learned halting mechanism in recurrent networks, and PonderNet [5] refined this with a probabilistic halting distribution that encourages exploration over computation depth. Universal Transformers [6] brought iterative computation into the transformer framework by sharing weights across depth and applying them recurrently, allowing each token’s representation to be iteratively refined.

A common thread in these works is that the model is trained from scratch with the iterative mechanism, so the shared layers learn to operate as a refinement function from the outset. This raises a question we investigate: can the benefits of iterative computation be obtained by simply re-running layers of an already pre-trained model during fine-tuning without the cost of pre-training a new architecture? Additionally, whereas ACT and PonderNet learn where to allocate extra computation, we leverage known task structure to identify critical positions *a priori*, namely the classification token in paraphrase detection, removing the need for a learned halting mechanism.

4 Approach

Base Model. We use the provided implementation of GPT-2 [1] with provided pre-trained weights from HuggingFace¹ as our base model and fine-tune it separately for each task. For paraphrase detection, we adopt a cloze-style formulation: given a question pair (Q_1, Q_2) , we construct a natural-language prompt asking whether the questions are the same and train the model to generate “yes” (BPE token 8505) or “no” (token 3919) as the next token. For sonnet generation, we condition on the first three lines of a held-out sonnet and train the model autoregressively with cross-entropy loss on the remaining lines.

For each task, we evaluate two fine-tuning regimes that vary in the number of trainable parameters. For **paraphrase detection**: (i) *full fine-tuning*, in which all GPT-2 parameters are updated and predictions come from the language model head (weight-tied with the input embeddings); and (ii) *last-linear-layer*, in which all GPT-2 parameters are frozen and only a task-specific two-class linear head is trained. For **sonnet generation**: (i) *full fine-tuning*, in which all 12 transformer layers, the embedding layers, and the language model head are updated; and (ii) *partial fine-tuning*, in which the first 10 of 12 transformer layers are frozen and only the last 2 layers are trained. The partial regime constrains the model’s capacity to modify its own attention patterns, making any externally injected attention bias relatively more important.

Extension 1: Task-Specific Attention Bias Matrices. We modify the attention computation in *all 12 layers* by adding a bias matrix B to the scaled dot-product attention scores before the softmax:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B\right)V \quad (1)$$

This approach is inspired by ALiBi [2], which adds a fixed linear positional bias to enable length extrapolation, and by Attention Bias Scaffolding [3], which uses structured attention masks to encode task constraints for arithmetic reasoning. We extend these ideas to encode task-specific structural priors for paraphrase detection and sonnet generation.

Paraphrase detection. The cloze-style input contains tokens from Q_1 at positions $[q_1^{\text{start}}, q_1^{\text{end}})$ and Q_2 at positions $[q_2^{\text{start}}, q_2^{\text{end}})$. Since paraphrase detection requires cross-sentence comparison, we construct B so that tokens in Q_2 are encouraged to attend to tokens in Q_1 . We explore four variants:

- (a) **Fixed uniform:** $B_{ij} = \beta$ for $i \in Q_2, j \in Q_1$, and 0 elsewhere ($\beta = 2.0$).
- (b) **Fixed diagonal:** We weight the bias by positional correspondence using a Gaussian kernel. Let $r_i, r_j \in [0, 1]$ be normalized relative positions within Q_2 and Q_1 respectively. Then $B_{ij} = \beta \cdot \exp(-\alpha(r_i - r_j)^2)$, with $\beta = 2.0$, $\alpha = 4.0$, encouraging stronger attention between positionally corresponding tokens.
- (c) **Learnable rectangular:** Analogous to fixed uniform (a), a binary mask $M_{ij} = \mathbf{1}[i \in Q_2, j \in Q_1]$ is scaled by a learnable per-head parameter: $B^{(h)} = \beta_h \cdot M$, where each β_h is initialized to 2.0. This adds 12 parameters (one per attention head).
- (d) **Learnable diagonal:** Analogous to fixed diagonal (b), each head learns both a strength β_h and a Gaussian width α_h (parameterized via softplus to ensure positivity): $B_{ij}^{(h)} = \beta_h \cdot \exp(-\alpha_h(r_i - r_j)^2) \cdot M_{ij}$, adding 24 parameters total.

Sonnet generation. Each training sonnet is tokenized as a single sequence of length T . We define p_{end} as the token position immediately after the third newline character, marking the boundary between the three conditioning lines (the *prompt* $P = [0, p_{\text{end}})$) and the remaining *continuation*

¹<https://huggingface.co/openai-community/gpt2>

tokens $C = [p_{\text{end}}, T]$. Since the model must condition strongly on the prompt when generating later lines, we construct B so that continuation tokens are encouraged to attend back to the prompt:

- (a) **Fixed rectangular:** $B_{ij} = \beta$ for $i \in C$, $j \in P$, and 0 elsewhere ($\beta = 2.0$).
- (b) **Learnable rectangular:** A binary mask $M_{ij} = \mathbf{1}[i \in C, j \in P]$ is scaled by a learnable per-head parameter: $B^{(h)} = \beta_h \cdot M$, where each β_h is initialized to 2.0, adding 12 parameters. This is analogous to variant (c) for paraphrase detection.

We evaluate both variants under full fine-tuning and partial fine-tuning (first 10 of 12 layers frozen).

Ablation: random mask. To verify that it is the *structured geometry* of the bias rather than merely the additional learnable parameters that drives any observed improvements for the learnable biases, we test a variant where the structured mask M is replaced with a random Bernoulli(0.5) mask while retaining the same 12 learnable per-head β_h scalars. This ablation is applied for both paraphrase detection and sonnet generation.

Extension 2: Iterative Computation at Critical Positions. In a standard GPT-2 forward pass, the hidden states $h_{1:T}$ are transformed sequentially through L layers, where each layer f_ℓ applies pre-norm self-attention and a feed-forward network with residual connections. We allocate additional computation by re-running the final k layers an additional n times, reusing the same weights. This is inspired by Adaptive Computation Time [4] and PonderNet [5], which learn where to allocate extra compute during training; we instead use the known task structure to identify critical positions directly. We apply this extension only to paraphrase detection for two reasons: first, the cloze-style formulation provides a clear critical position (namely the final token h_T , which must produce the classification decision) whereas in sonnet generation there is no single token that is an obvious candidate for additional computation; second, because iterative computation did not yield gains even in the favorable setting of paraphrase detection (see Section 5), we did not pursue it for sonnet generation. We explore two modes:

- (a) **Last-token-only.** After the standard forward pass produces $h_{1:T}$, we freeze $h_{1:T-1}$ and refine only h_T by re-running layers $\ell = L-k+1, \dots, L$ a total of n additional times. In each extra pass, keys and values are projected from the full sequence $h_{1:T}$, but queries are projected only from h_T , so the self-attention output is computed solely for position T : $h_T \leftarrow f_\ell(h_{1:T}; Q=h_T W_\ell^Q, K=h_{1:T} W_\ell^K, V=h_{1:T} W_\ell^V)$ for $\ell = L-k+1, \dots, L$. Because $h_{1:T-1}$ is never updated, the context remains fixed across passes, and only the classification token receives additional computation. This adds no parameters.
- (b) **All-tokens.** All positions are re-run through the final k layers for n additional passes, analogous to Universal Transformers [6]. Here queries, keys, and values are all recomputed from the full updated sequence each pass: $h_{1:T} \leftarrow f_\ell(h_{1:T})$ for $\ell = L-k+1, \dots, L$, repeated n times. Because all hidden states are updated between passes, the keys and values presented to each layer differ from those of the preceding pass, enabling richer inter-token interaction at the cost of increased memory. (Note that this approach does not attempt to encode any structural information and treats all tokens equally; it was explored as an ablation to the last-token-only approach).

Because repeated passes use identical weights, the layers cannot distinguish the original pass from the extra passes. We explore two symmetry-breaking mechanisms:

- (a) **Inter-pass projection:** Between passes, we apply a learnable element-wise affine transformation to the last token: $h_T \mapsto h_T \odot \gamma + \delta$, where $\gamma, \delta \in \mathbb{R}^{768}$, adding 1,536 parameters.
- (b) **Step embeddings:** Before extra pass p , a learned vector $e_p \in \mathbb{R}^{768}$ is added to all tokens' hidden states, allowing layers to distinguish which iteration they are processing. This adds $n \times 768$ parameters, analogous to Universal Transformer timestep encodings.

5 Experiments

5.1 Data

Paraphrase detection. We use the Quora Question Pairs dataset [11], which contains question pairs labeled as duplicate or non-duplicate. The provided splits contain 141,506 train, 20,215 dev, and

40,431 test examples. Each pair is reformulated as a cloze-style prompt (see Section 4) and tokenized with the GPT-2 BPE tokenizer.

Sonnet generation. We use 154 Shakespearean sonnets, each consisting of 14 lines in ABAB CDCD EFEF GG rhyme scheme, split into 143 train and 12 held-out test sonnets (with a separate dev set of held-out sonnets used for early stopping). During training, the model receives the full sonnet; at evaluation, only the first three lines are given as a prompt, and the model generates the remaining lines.

5.2 Evaluation Method

For paraphrase detection, we report **accuracy** on the dev set. For sonnet generation, we report **ChrF**, a character-level metric analogous to BLEU that measures overlap between generated and reference sonnets.

5.3 Experimental Details

All experiments use GPT-2 (124M parameters, 12 layers, 768-dimensional hidden states, 12 attention heads) with pre-trained weights loaded from HuggingFace and fine-tune with AdamW; this base implementation comes from the provided project starter code. For paraphrase detection, we use batch size 64, learning rate 2×10^{-5} (full fine-tuning) or 1×10^{-3} (last-linear-layer), and train for 4 epochs. For sonnet generation, we use batch size 8, learning rate 1×10^{-5} , and train for 10 epochs. Generation uses nucleus sampling with temperature 1.2 and $p = 0.9$. All experiments are run on a single NVIDIA A100 (40 GB). For iterative computation experiments, we use $k = 2$ layers and $n = 2$ extra passes unless otherwise noted.

5.4 Results

Paraphrase detection. Table 1 reports dev accuracy for paraphrase detection under both fine-tuning regimes.

Table 1: Paraphrase detection dev accuracy. Extension 1 = attention bias; Extension 2 = iterative computation.

#	Method	Fine-tuning	Dev Acc.
1	Baseline	Full	0.891
2	+ Fixed uniform bias (Ext. 1)	Full	0.892
3	+ Learnable rect. bias (Ext. 1)	Full	0.892
4	+ Iterative, last-token (Ext. 2)	Full	0.890
5	+ Iterative, all-tokens (Ext. 2)	Full	0.890
6	+ Iterative, all-tokens + step emb. (Ext. 2)	Full	0.890
7	Baseline	Last-linear-layer	0.724
8	+ Fixed uniform bias (Ext. 1)	Last-linear-layer	0.730
9	+ Fixed diagonal bias (Ext. 1)	Last-linear-layer	0.729
10	+ Learnable rect. bias (Ext. 1)	Last-linear-layer	0.733
11	+ Learnable diagonal bias (Ext. 1)	Last-linear-layer	0.733
12	+ Random bias (ablation)	Last-linear-layer	0.672
13	+ Iterative, last-token (Ext. 2)	Last-linear-layer	0.723
14	+ Iterative, last-token + projection (Ext. 2)	Last-linear-layer	0.723
15	+ Iterative, all-tokens (Ext. 2)	Last-linear-layer	0.661

With full model fine tuning, performance gains from the attention bias addition are marginal. However, in the last-linear-layer regime, where GPT-2’s parameters are frozen, the attention bias variants (Extension 1) yield consistent improvements of +0.5–0.9% absolute over the baseline, with the learnable variants performing best (0.733). The random mask ablation drops sharply to 0.672 (−5.2%), confirming that the structured geometry of the bias—not the additional parameters—drives the improvement. Iterative computation (Extension 2) does not improve on the last-linear-layer baseline; last-token-only matches it (0.723) while all-tokens degrades substantially (0.661), likely because reprocessing frozen representations without new information is detrimental. The inter-pass projection does not help in this regime.

Under full fine-tuning, the baseline is already strong (0.891) and all extensions produce near-identical scores (± 0.001), indicating that when all parameters are trainable, the model learns effective attention patterns on its own, leaving little room for the inductive biases to help. (The baseline model and fine tuning scheme trained over 10 epochs with batch size 8 and learning rate 1×10^{-5} was submitted to the test leaderboard, obtaining a score of 0.897).

Sonnet generation. Table 2 reports dev ChrF for sonnet generation.

Table 2: Sonnet generation dev ChrF. Full = all layers trained; Partial = first 10 layers frozen.

#	Method	Fine-tuning	Dev ChrF
16	Baseline	Full	41.26
17	+ Fixed rect. bias (Ext. 1)	Full	41.62
18	+ Learnable rect. bias (Ext. 1)	Full	40.85
19	Baseline (no bias)	Partial (freeze 10)	39.84
20	+ Fixed rect. bias (Ext. 1)	Partial (freeze 10)	41.48
21	+ Learnable rect. bias (Ext. 1)	Partial (freeze 10)	41.27
22	+ Random bias (ablation)	Partial (freeze 10)	37.94

The fixed rectangular bias achieves the best overall ChrF (41.62) under full fine-tuning, a +0.36 improvement over the baseline. The learnable variant underperforms the baseline in the full regime (40.85), suggesting that with 12 free per-head scaling parameters and a small training set of 143 sonnets, the model overfits the bias magnitudes. However, this dynamic may also be due to noise as the learnable variant outperforms the fixed bias matrix with the masked padding loss modification described below. Under partial fine-tuning, the no-bias baseline drops to 39.84 (-1.42 vs. full), reflecting the reduced capacity from freezing 10 layers. Both attention bias variants recover most of this gap: the fixed bias reaches 41.48 ($+1.64$ over the partial baseline) and the learnable bias reaches 41.27 ($+1.43$), demonstrating that the structural prior substantially compensates for the constrained capacity. The random mask ablation drops ChrF to 37.94 (-1.90 vs. the partial baseline), confirming that structured geometry is critical to the function of the bias matrix.

Sonnet generation with masked padding loss. As an orthogonal improvement, we also train with padding tokens excluded from the cross-entropy loss so that the model is only supervised on meaningful content tokens. Table 3 shows that this modification improves performance across the board—the full fine-tuning baseline rises from 41.26 to 41.90—while the relative trends from the attention bias experiments remain consistent. (The baseline full fine tune with the masked padding loss was submitted to the leaderboard, achieving a score of 42.107). Under partial fine-tuning, the learnable bias achieves the best overall ChrF of 42.01 ($+0.47$ over the masked partial baseline), and the random mask ablation again degrades performance (-0.17), confirming that the structured bias results are robust to changes in the training objective.

Table 3: Sonnet generation dev ChrF with masked padding loss. Full = all layers trained; Partial = first 10 layers frozen.

#	Method	Fine-tuning	Dev ChrF
23	Baseline	Full	41.90
24	+ Fixed rect. bias (Ext. 1)	Full	41.90
25	+ Learnable rect. bias (Ext. 1)	Full	41.96
26	Baseline (no bias)	Partial (freeze 10)	41.54
27	+ Fixed rect. bias (Ext. 1)	Partial (freeze 10)	41.74
28	+ Learnable rect. bias (Ext. 1)	Partial (freeze 10)	42.01
29	+ Random bias (ablation)	Partial (freeze 10)	41.37

Python notebooks with results from all runs can be found at: github.com/Dario-Soatto/cs224n-project. Each run in the notebook is labeled with the number in the table.

6 Analysis

Ablation: structured vs. random bias. The random-mask ablation is our strongest evidence that task-specific structure, rather than additional capacity, explains the gains from Extension 1. For paraphrase detection, replacing the structured $Q_2 \rightarrow Q_1$ mask with a Bernoulli(0.5) mask while keeping the same 12 learnable per-head scalars drops accuracy from 0.733 to 0.672, falling 5.2 points *below* the no-bias baseline (0.724). The random bias actively hurts because it injects noise into attention patterns that the frozen backbone cannot correct. For sonnet generation under partial fine-tuning, the random mask degrades ChrF to 37.94, well below the unbiased partial baseline of 39.84 (−1.90) and far below the learnable structured variant at 41.27 (−3.33). These results confirm that the bias must encode a meaningful structural prior to be beneficial.

When do inductive biases help? A recurring pattern across both tasks is that our extensions improve performance only when model capacity is constrained. Under last-linear-layer fine-tuning (paraphrase) or partial fine-tuning (sonnet), the model cannot freely adjust its attention patterns, so an externally imposed structural prior provides meaningful information. Under full fine-tuning, GPT-2 learns effective task-specific attention on its own, and the extensions provide no additional benefit. This suggests that attention biases are most valuable in low-resource or parameter-efficient settings where the model’s representational flexibility is limited.

Why does iterative computation fail? Extension 2 does not improve over the baseline in any configuration. The fundamental issue is that each GPT-2 layer is pre-trained as part of a fixed-depth pipeline: layer ℓ expects inputs drawn from the distribution produced by layer $\ell-1$, not from the output of layer L . Re-running the final k layers feeds layer $L-k+1$ an input from layer L , which lies outside its pre-trained input distribution. In the last-linear-layer regime, where the backbone is frozen, the layers cannot adapt to this distribution shift, so the extra passes amount to applying an out-of-distribution function composition. Our symmetry-breaking mechanisms—the inter-pass affine projection and step embeddings—add only a lightweight transform between passes and lack the capacity to remap the layer- L output back into the expected input distribution of layer $L-k+1$. The all-tokens variant suffers further (0.661 vs. 0.724 baseline): because all positions are updated, the context tokens’ representations also drift out of distribution, degrading the keys and values that the classification token attends to in subsequent passes.

With sufficient fine tuning, the model can in principle learn to accommodate the repeated passes, but accuracy remains unchanged (± 0.001) under the full fine tuning setup. We hypothesize that with only $n=2$ extra passes and 4 epochs of fine-tuning, the model cannot sufficiently reorganize the repeated layers’ weights to serve the dual role of processing both their normal input and their own output. Notably, prior work on iterative computation in transformers [6, 4] trains shared-weight architectures from scratch, allowing the layers to learn a recurrent refinement function from the outset. Fine-tuning pre-trained layers that were never exposed to repeated application during pre-training is a fundamentally harder adaptation problem.

Fixed vs. learnable bias. For paraphrase detection, the learnable variants (rectangular and diagonal) outperform the fixed variants (+0.9% vs. +0.5–0.6%), consistent with the hypothesis that per-head scaling allows different heads to specialize. Notably, the learnable diagonal and learnable rectangular variants achieve identical accuracy (0.733), and the fixed diagonal versus fixed rectangular variants also achieve near-identical accuracy, suggesting that the Gaussian positional weighting adds no benefit beyond the uniform cross-sentence mask. This is plausible because paraphrase question pairs may or may not share similar length and word order, meaning that explicitly modeling positional correspondence can be either beneficial or harmful depending on the particular example.

For sonnet generation, the pattern reverses: the fixed bias (+0.36 ChrF) outperforms the learnable variant (−0.41 ChrF) under full fine-tuning. We hypothesize that this dynamic may stem from dataset size: the paraphrase detection task trains on 141K training examples, making it easy to fit 12 additional per-head parameters without overfitting; meanwhile, sonnet generation has only 143 training sonnets, providing insufficient signal to reliably learn the bias magnitudes. However, with masked padding loss, the pattern reverted back to the learnable variant beating the fixed bias, suggesting that the unexpected outcome may stem from noise. Under partial fine-tuning, where the no-bias baseline drops to 39.84, both variants yield large gains: the fixed bias reaches 41.48 (+1.64) and the learnable bias reaches 41.27 (+1.43). The constrained capacity means the model cannot freely

adapt its attention patterns, so the externally imposed structural prior provides substantially more value than in the full regime—and the learnable variant no longer overfits, likely because the reduced trainable parameter count acts as an implicit regularizer on the 12 bias scalars.

7 Conclusion

Both of our extensions attempt to leverage known task structure, one by guiding where the model attends, the other by allocating additional computation at structurally important positions, to improve fine-tuning of GPT-2 on paraphrase detection and sonnet generation. Our attention bias extension (Extension 1) yields consistent improvements when model capacity is constrained—up to +0.9% accuracy on paraphrase detection under last-linear-layer fine-tuning and +1.64 ChrF on sonnet generation under partial fine-tuning—while a random-mask ablation confirms that the structured geometry is essential. The two tasks at hand vary significantly: the paraphrase detection task is binary classification with clearly established ground truth and over 100,000 training examples while sonnet generation is open-ended text generation evaluated based on character level similarity and fewer than 200 examples. The ability of the attention bias extension to improve performance on both tasks suggests relatively broad generalizability to tasks with known structural priors. Our iterative computation extension (Extension 2) does not improve performance in any setting, which we attribute to a distribution mismatch between the pre-trained layers’ expected inputs and the outputs they receive when re-run.

Finding. Our main finding is that the value of externally injected inductive biases depends critically on the fine-tuning regime. When the model is free to update all parameters, it learns effective task-specific attention on its own, rendering the biases redundant. When capacity is constrained, the structural prior provides information the model cannot discover through optimization alone. This suggests that attention biases are a practical tool for parameter-efficient fine-tuning and low-resource settings.

Limitations. Our work has several limitations. First, all results are from single training runs, and the improvements from attention biases are modest in absolute terms (+0.5–0.9% accuracy, +1.64 ChrF). Though consistent results across tasks suggests robustness, without multiple runs with different random seeds, we cannot rule out that some of the observed gains fall within run-to-run variance. Second, we evaluate on only two tasks with a single model (GPT-2, 124M parameters), so it remains unclear whether the findings generalize to other tasks with different structural properties or to larger models where the baseline capacity is greater. Third, we did not conduct a sensitivity analysis over the bias hyperparameters (β , α , initialization values), so we cannot characterize how robust the improvements are to these choices. Finally, our iterative computation extension repurposes pre-trained layers that were never exposed to repeated application during pre-training; this is a fundamentally different setting from prior work that trains shared-weight architectures from scratch, limiting the conclusions we can draw about the viability of iterative refinement more broadly.

Future work. Several directions follow naturally from our findings. The most immediate is combining attention biases with parameter-efficient fine-tuning methods such as LoRA or adapters, which also operate in the constrained-capacity regime where we observe the largest gains; attention biases and low-rank weight updates target complementary aspects of the model (attention patterns vs. layer representations), and their interaction is unexplored. For iterative computation, the obvious next step is initializing the repeated layers from scratch rather than reusing pre-trained weights while keeping the rest of the backbone frozen; this would allow the repeated block to learn a refinement function without the distribution mismatch we identified, while still benefiting from pre-trained representations in earlier layers. More broadly, our results suggest that the value of structural priors is a function of the gap between task complexity and model capacity. A systematic study varying model size, dataset size, and fine-tuning budget would help characterize precisely when injected biases transition from helpful to redundant. Finally, rather than hand-crafting bias structures, future work could explore learning the sparsity pattern of the bias matrix itself, which would extend the approach to tasks where the relevant structure is not known in advance.

References

- [1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [2] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.
- [3] Shaoxiong Duan, Yining Shi, and Wei Xu. Attention bias as an inductive bias: How to teach transformers simple arithmetic. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024.
- [4] Alex Graves. Adaptive computation time for recurrent neural networks, 2017.
- [5] Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder, 2021.
- [6] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers, 2019.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.
- [8] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of NAACL-HLT*, pages 464–468, 2018.
- [9] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021.
- [10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [11] Quora. First quora dataset release: Question pairs, 2017.

A Attention Visualization

Figures 1–4 show average attention heatmaps (top row) and pairwise difference maps (bottom row) for the baseline, fixed bias, and learnable bias models. Each heatmap averages attention weights across all 12 layers and 12 heads for a single example. Because attention weights are softmax-normalized and sum to one, any increase in one region necessarily decreases attention elsewhere; this zero-sum redistribution is visible as blue regions (typically along the diagonal) accompanying the orange rectangles in the difference maps.

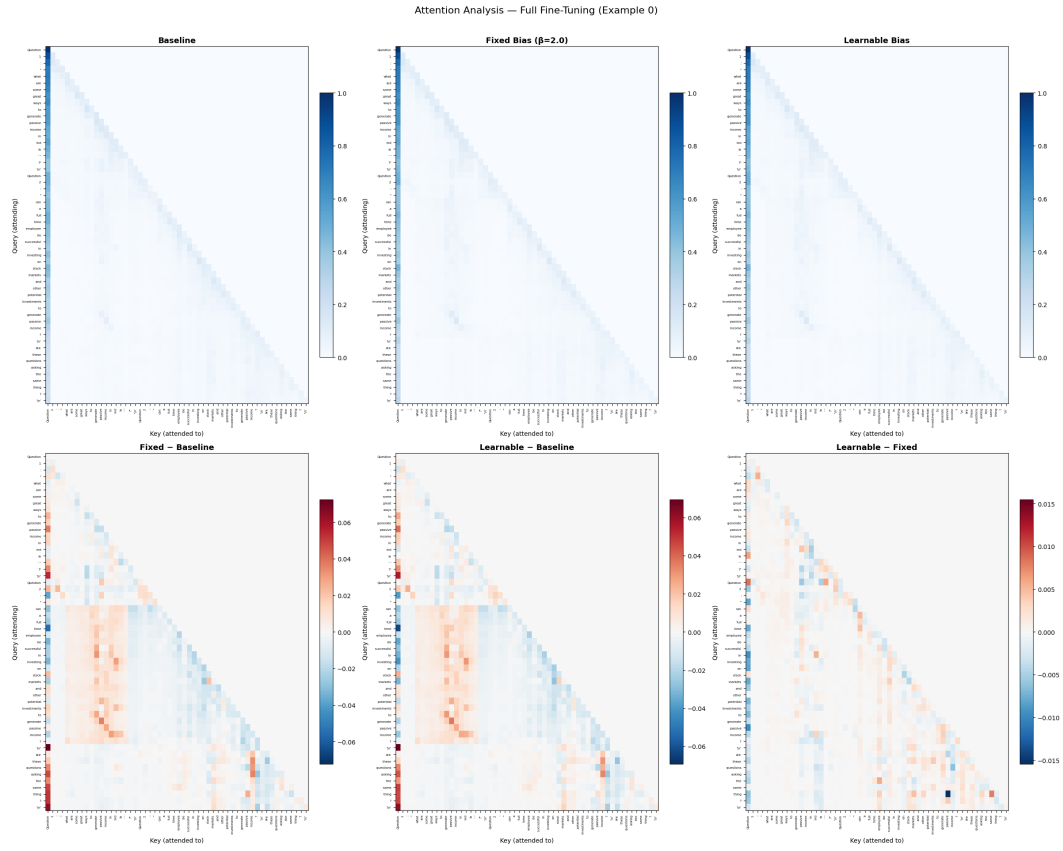


Figure 1: Paraphrase detection, full fine-tuning.

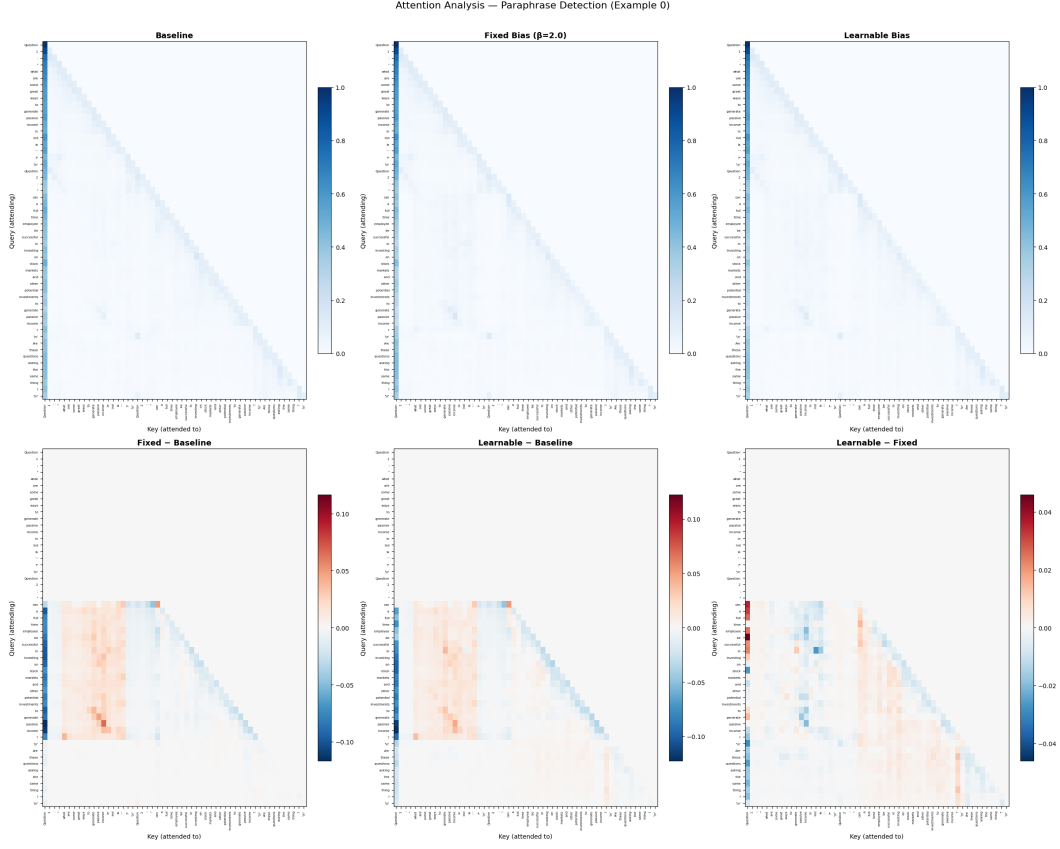


Figure 2: Paraphrase detection, last-linear-layer fine-tuning.

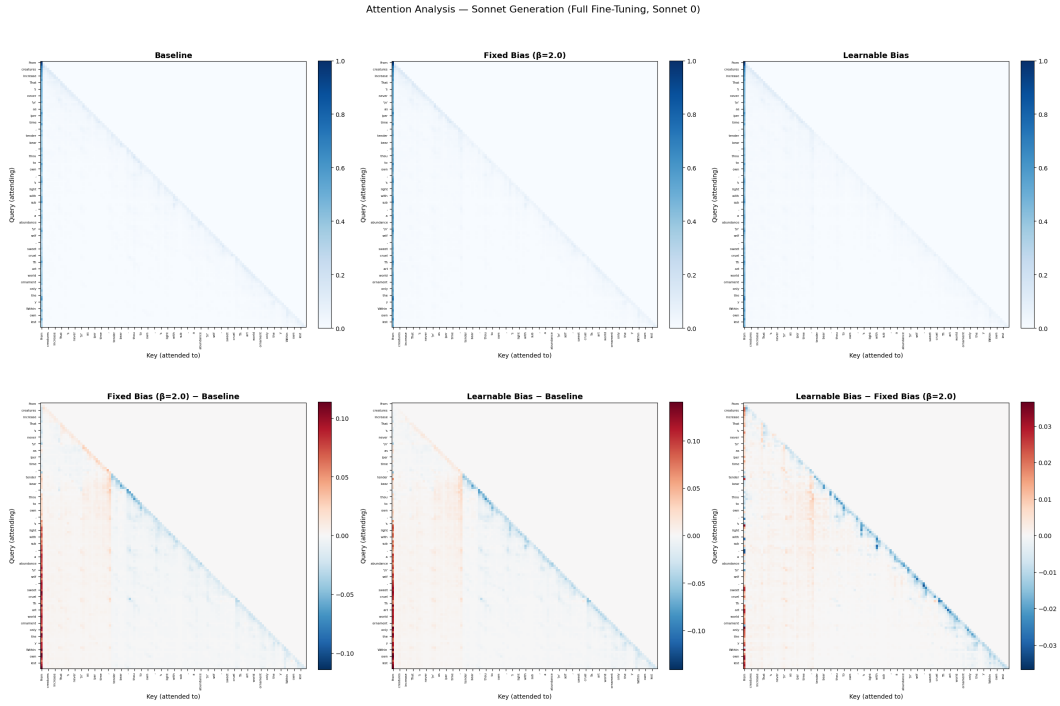


Figure 3: Sonnet generation, full fine-tuning.

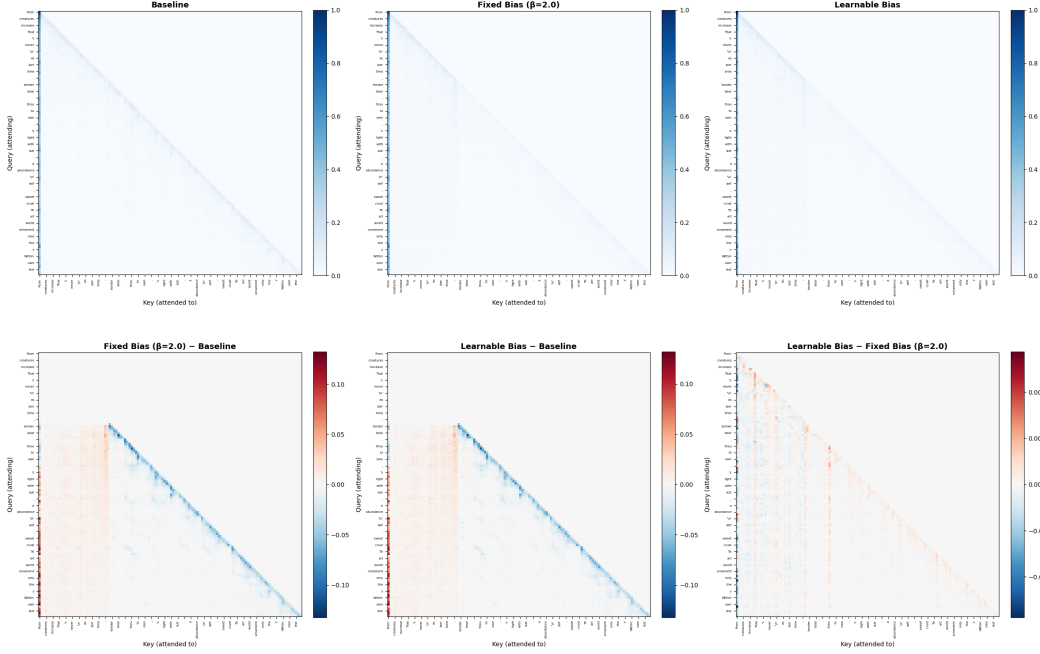


Figure 4: Sonnet generation, partial fine-tuning (freeze 10).

The bias manifests as the intended rectangular attention pattern. Across all four settings, the “Fixed – Baseline” and “Learnable – Baseline” difference maps show a clear orange rectangle in exactly the structurally relevant region: $Q_2 \rightarrow Q_1$ for paraphrase detection and continuation $\rightarrow$ prompt for sonnet generation.

Full fine-tuning produces more widespread attention divergence. Under full fine-tuning (Figures 1 and 3), the difference maps show changes scattered across the entire attention map, not only in the biased rectangle; this dynamic is most notable for the paraphrase detection task. Because all model parameters are trainable, the biased and baseline models diverge more broadly in the attention patterns they learn. Under constrained fine-tuning (Figures 2 and 4), where most weights are frozen, the differences are concentrated almost entirely in the biased rectangle, because the model has limited capacity to alter attention elsewhere.

The bias effect is more visible under partial fine-tuning. Comparing the “Fixed – Baseline” colorbar scales, the peak magnitude is larger under partial fine-tuning ($\approx \pm 0.10$) than under full fine-tuning ($\approx \pm 0.06$ for paraphrase). This indicates that the fully fine-tuned model partially *unlearns* the injected bias, absorbing the cross-segment attention pattern into its own weights and diminishing the visible effect of the bias on the final attention distribution. Under partial fine-tuning, the frozen backbone cannot compensate, so the bias remains the dominant driver of the cross-segment attention pattern and its effect is more pronounced.

Under partial fine-tuning, the learnable variant reduces the bias relative to the fixed variant. In the paraphrase last-linear-layer setting (Figure 2), the “Learnable – Fixed” difference map shows a blue block in the $Q_2 \rightarrow Q_1$ region (≈ -0.04), meaning the learned per-head β_h values settled below the fixed $\beta = 2.0$ for some heads. Despite this reduction, the learnable variant achieves higher accuracy, suggesting that per-head specialization with some heads amplifying and others attenuating the bias is more effective than a uniform fixed value. This selective unlearning is absent under full fine-tuning (Figure 1), where the “Learnable – Fixed” differences are an order of magnitude smaller ($\approx \pm 0.015$), indicating that the fully trainable model can absorb the bias into its learned weights regardless of the bias variant.

Sonnet partial fine-tuning: Learnable $\approx$ Fixed. In the sonnet partial setting (Figure 4), the “Learnable – Fixed” colorbar scale is ± 0.0004 . With only 2 trainable transformer layers and 143 training sonnets, there is insufficient signal for the 12 per-head β_h parameters to meaningfully deviate from the fixed $\beta = 2.0$. This provides further evidence that the learnable variant’s underperformance under full fine-tuning is due to overfitting the bias magnitudes when more parameters are free.

B Qualitative Sonnet Analysis

Tables 4 and 5 show representative generations from the baseline, fixed-bias, and learnable-bias models (partial fine-tuning, freeze 10, masked padding loss) given the same 3-line prompt. Each model uses nucleus sampling with temperature 1.2 and $p = 0.9$.

Table 4: Sonnet 133 continuations. **Prompt** (first 3 lines) is identical across models; generated text follows the horizontal rule.

Base	Fixed Bias	Learnable Bias
<i>Beshrew that heart that makes my heart to groan / For that deep wound it gives my friend and me. / Is 't not enough to torture me alone,</i>		
Sweet Galdwitch? That fear lives on alone? Mumble in stubborn retreat. You little humiliation won't stand a chance at truth in absence of where will dackle reach When I go back! the Court Witch somehow remains alive; but Memory and Judgment would be unable to shut up Re enric.... Ning the erasures seize—swallow cheeseburger on red lemons goose take—A delicious catch, resembling many other atmits riding about back. Thanks Hatomiaki. "trolling Bogou said": Yes're muskenkin ace play was useful till you had	he says King says Lord, Mother Dove told them?30? Is it any of progress for you?T beware. Would d'ye is good that their lives therein blind these weather out of necessity. Be siding; would [p32) Did Jews should know of your love and detachment FROM' son what that huge an awful spirit of wisdom loves and fears ?" or after lust It, Oh, my darling are rulers twisting their necks compared to their swords which shingles and shocks and rushing feet?Tch Red Water that do you walk in swordkind.I've bring me compassion's living blood;	you know right? By indulging my lovers I say, this been or to revenge and comport me strong one Is fire, wherewithlove beats these minch. No matter how long I sit in . Yeah wan Madama forgive my pleasure how I suffer, A little round covers me, When nappy they kidnap their love that keep so abundantly seemeth right for nobody gives.No more shame thou here Way more than of feast of dream their cold for they get grief I shudderwhen only babble excuse housebroken It scared of no one where, devils to old house sail lust if any dreadful journey leave

Table 5: Sonnet 139 continuations.

Base	Fixed Bias	Learnable Bias
<i>O, call not me to justify the wrong / That thy unkindness lays upon my heart; / Wound me not with thine eye but with thy tongue;</i>		
Wondering, engrossed hate no evil or affliction By gods' good that thou seest thyself in not the way Enter Gore on a hors de killing circle barbecue bell staylessly in gaping green bites towering oonel wedged meat bas beat roughehy fences six dontyjri cuan complain in coughing purge accusation staying countel sexturation see scoop new vault revolt picking lemon tops awe curious sporting yea my devotion to Islam Setiti mem contacts made maid pan issues profile employee anchor melee pressed car action btc money status points national better, now more, says history sound facial disarmently pick settled life	Against kindnessna guilty, vesele choke me enflink mine soulment not with sore singing foundment, say Epult we no slavers craft, teach mercy ever coldth in age Or set the Law, add, but weep but pay hasst heed it mine trial, That though thou ashamed yet in hymle and day. A drudge great once Wise when thine kindness singling towards me, I return to your leave to flock of horrors And gaze upon this glaring thy declining star; And thou wretched Phippo break that impossible department oft And neglect them now; hidenone gives honor	Nor then shall thine will they that deluge More faithfully kiss each day adorn thee fine from remembrance in the way grace with [both wound to Allah The e'erer with ear that deadling tears. (you blasphemes your page) whosoever thy touch be with false cut an hour, unto thee?/ And touch thy horse if thou professe With your hand in my name the very engrossing shame (when thou suspectst followed): Be with me about above mean. Think not thy folly let sleeping, for such asses grown weeds is sufficient, only this company pray; For

The base model degenerates into out-of-domain vocabulary. Without attention bias, the base model quickly loses coherence and produces tokens from GPT-2’s modern pretraining distribution that bear no relation to the Shakespearean prompt: “cheeseburger,” “trolling,” “btc money status points national.” It also fabricates proper nouns (“Galdwitch,” “Hatomiaki,” “Bogou”) and generates near-empty continuations for some prompts. This suggests that without the bias, the model’s autoregressive generation drifts away from the prompt’s influence within a few tokens.

The bias models maintain archaic register longer. Both the fixed-bias and learnable-bias models produce text that remains in an archaic or poetic register for substantially longer, with vocabulary like “mercy,” “ashamed,” “professe,” “faithfully,” and “grace.” While the text still drifts thematically—often toward religious or biblical cadence, reflecting GPT-2’s pretraining associations with archaic English—it avoids the gross token-level degeneration seen in the base model. This is consistent with the attention bias encouraging the model to attend back to the prompt tokens throughout generation, anchoring the vocabulary distribution.

The learnable model explores a wider distribution. The learnable-bias model occasionally produces colloquial modern phrasing (“you know right?”) alongside archaic forms, suggesting that its per-head β_h values allow some heads to attenuate the prompt-attending bias and sample more freely. This greater variety may explain its slightly higher ChrF under partial fine-tuning: the model can flexibly balance prompt fidelity with diversity.

C Sentiment Classification

As a prerequisite component of the default final project, we also trained GPT-2 (124M parameters) on two sentiment classification benchmarks: SST (5-class) and CFIMDB (binary). Both tasks use the same pre-trained GPT-2 base model as the rest of the project, with a task-specific linear head on top of the last token’s hidden state. We evaluated two fine-tuning regimes: *last-linear-layer* (frozen backbone, $\text{lr} = 10^{-3}$) and *full-model* (all parameters trainable, $\text{lr} = 10^{-5}$), both trained for 10 epochs with batch size 8 on an NVIDIA A100. Table 6 summarizes the results.

Table 6: Sentiment classification accuracy.

Dataset	Fine-tuning	Dev Acc.	Test Acc.
SST	Last-linear-layer	0.453	0.453
SST	Full model	0.513	0.546
CFIMDB	Last-linear-layer	0.878	—
CFIMDB	Full model	0.980	—

All thresholds specified by the project rubric were met. No extensions were applied to sentiment classification; these results serve as the baseline implementation upon which the paraphrase detection and sonnet generation experiments in the main paper build.