LATENTWIRE: CROSS-MODEL COMMUNICATION VIA SOFT TOKENS

Sujeeth Jinesh [1] **Thierry Tambe** [1]

ABSTRACT

We present LatentWire, a method for cross-model communication through learned soft tokens that bypasses autoregressive text generation. A Perceiver Resampler bridge transforms hidden states from a sender LLM (Llama 3.1 8B) into soft tokens conditioning a receiver (Mistral 7B). On classification benchmarks, LatentWire achieves **90.7% average accuracy** across SST-2, AG News, and TREC with 32 soft tokens—outperforming zero-shot baselines by +4.1pp, +19.6pp, and +37.3pp respectively—while providing **4.66× speedup** over text-relay. Linear probes validate that task-relevant information is extractable from sender representations (95% on TREC). However, we identify critical limitations: reasoning tasks fail fundamentally (CommonsenseQA below random chance), and extreme compression (8 tokens) causes high variance. These results demonstrate that continuous representations enable faster, more effective cross-model communication than text for classification, while highlighting the need for hybrid architectures to handle reasoning.

1 INTRODUCTION

Large language models (LLMs) have emerged as powerful tools for natural language understanding and generation (Vaswani et al., 2017; Touvron et al., 2023; Jiang et al., 2023a). However, the dominant paradigm for combining multiple LLMs involves sequential text generation: one model produces text that another model consumes. This approach incurs substantial latency due to autoregressive decoding and may lose information through the discretization bottleneck of natural language.

We propose **LatentWire**, a method that enables direct communication between heterogeneous LLMs through learned soft tokens. Rather than having a sender model generate text for a receiver model to process, LatentWire transforms the sender's internal representations into a small set of continuous embeddings (soft tokens) that directly condition the receiver model's inference. On classification benchmarks, this approach:

1. **Reduces latency significantly**: The sender model only performs a single forward pass, achieving 4.66× speedup compared to text-relay approaches.
2. **Preserves task-relevant information**: Soft tokens encode features better than text serialization, achieving +19.6pp over zero-shot on AG News and +37.3pp on TREC.
3. **Enables cross-model collaboration**: The bridge achieves 90.7% average accuracy across three tasks, substantially outperforming individual models.

Our key contributions are:

- A bridge architecture based on Perceiver Resampler (Jaegle et al., 2021; Alayrac et al., 2022) that achieves 90.7% average accuracy (SST-2: 93.7%, AG News: 90.7%, TREC: 87.9%) while being 4.66× faster than text-relay.
- Comprehensive evaluation showing the bridge substantially outperforms zero-shot baselines (+4.1pp SST-2, +19.6pp AG News, +37.3pp TREC) and dramatically outperforms text-relay (+52–90pp).
- Linear probe analysis showing 95% accuracy on TREC from sender hidden states, validating that task-relevant information is present and extractable.
- Identification of critical limitations: reasoning tasks fail fundamentally (CommonsenseQA below random), and extreme compression (8 tokens) causes high variance on complex tasks.

2 RELATED WORK

Soft Prompts and Prompt Tuning Prompt tuning (Lester et al., 2021) and prefix tuning (Li & Liang, 2021) demonstrated that freezing LLM weights while learning continuous "soft" prompt embeddings can match full fine-tuning performance. Our work extends this paradigm from

[1]Stanford University, Stanford, CA, USA. Correspondence to: Sujeeth Jinesh <sujinesh@stanford.edu>.

Proceedings of the 8^{th} MLSys Conference, Santa Clara, CA, USA, 2025.

single-model adaptation to cross-model communication, using soft tokens as an interlingua between heterogeneous models.

Perceiver and Vision-Language Bridging The Perceiver (Jaegle et al., 2021) introduced cross-attention to map arbitrary-length inputs to a fixed-size latent array. BLIP-2 (Li et al., 2023) and Flamingo (Alayrac et al., 2022) applied similar principles to bridge frozen image encoders and LLMs through learned query tokens. Our bridge architecture adapts this design for language-to-language transfer between heterogeneous models.

Model Stitching Model stitching (Bansal et al., 2021; Pan et al., 2023) connects layers from different networks using learned transformations. Unlike these *offline* methods that modify weights before deployment, LatentWire enables *runtime* communication through soft tokens during inference, supporting dynamic, input-dependent information flow.

Prompt Compression Methods like LLMLingua (Jiang et al., 2023b) compress prompts by removing tokens, while ICAE (Ge et al., 2024) and 500xCompressor (Li et al., 2024) learn dense embeddings. Our work focuses on cross-model communication rather than single-model compression.

Knowledge Distillation Knowledge distillation (Hinton et al., 2015) transfers knowledge from a teacher to a student model, typically through soft targets or intermediate representations. Recent work extends this to LLMs (Xu et al., 2024; Gu et al., 2024), but focuses on training smaller models to mimic larger ones. In contrast, LatentWire enables runtime collaboration between frozen models without modifying either model's weights.

Continuous Latent Reasoning Recent work explores reasoning in continuous latent spaces rather than through explicit text tokens. Coconut (Hao et al., 2024) trains models to reason via continuous "thought tokens" without verbalization, achieving improved efficiency on reasoning tasks. Our soft tokens serve a different purpose—cross-model communication rather than internal reasoning—but share the insight that continuous representations can encode information more efficiently than discrete text.

3 METHOD

3.1 Problem Formulation

We formalize the cross-model communication problem as follows. Let $\mathcal{S}$ and $\mathcal{R}$ denote a sender and receiver LLM respectively, with potentially different architectures, tokenizers, and training distributions. Given input text x, we seek a communication protocol that enables $\mathcal{R}$ to perform a downstream task using information extracted from $\mathcal{S}$'s processing of x.

Desiderata An ideal cross-model communication mechanism should satisfy:

1. **Efficiency**: Communication should be faster than text generation ($O(1)$ vs $O(L)$ autoregressive steps).
2. **Fidelity**: Task-relevant information should be preserved through the channel.
3. **Modularity**: Both models remain frozen; only the communication channel is learned.
4. **Compression**: The transmitted representation should be compact ($M \ll L$ tokens).

Formal Setup Let $\mathbf{h}_{\mathcal{S}}^{(\ell)} \in \mathbb{R}^{L \times d_{\mathcal{S}}}$ denote the hidden states from layer ℓ of the sender, where L is the sequence length and $d_{\mathcal{S}}$ is the hidden dimension. We seek a bridge function $f_\theta : \mathbb{R}^{L \times d_{\mathcal{S}}} \to \mathbb{R}^{M \times d_{\mathcal{R}}}$ that produces soft tokens $\mathbf{z} = f_\theta(\mathbf{h}_{\mathcal{S}}^{(\ell)})$ satisfying:

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \; p_{\mathcal{R}}(y | \mathbf{z}, \mathbf{x}_{\text{prompt}}) \tag{1}$$

where y is the correct task output and $\mathbf{x}_{\text{prompt}}$ is an optional task-specific prompt.

Key Challenge: Representation Mismatch The sender and receiver occupy different representation spaces. Even when hidden dimensions match ($d_{\mathcal{S}} = d_{\mathcal{R}} = 4096$ for Llama and Mistral), the geometric structure differs due to:

- **Vocabulary**: Llama (128K tokens) vs. Mistral (32K tokens)
- **Positional encoding**: Different RoPE base frequencies
- **Attention**: Grouped-query (Llama) vs. sliding window (Mistral)
- **Statistics**: Hidden state magnitude differs by $\sim 5\times$

A naive linear projection fails because it assumes isomorphic spaces. The bridge must learn a *semantic translation*, not merely a coordinate transformation. Figure 1 illustrates the overall pipeline.

3.2 Bridge Architecture

Our bridge uses a Perceiver Resampler design:

1. **Input Projection**: Linear projection from sender hidden dimension to bridge internal dimension: $\mathbf{h}' = \mathbf{W}_{\text{in}} \mathbf{h}_{\mathcal{S}}$, where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d_{\mathcal{S}} \times d}$.

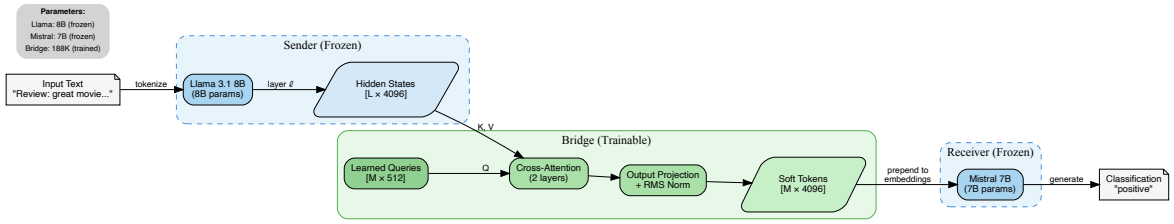

Figure 1. LatentWire architecture. Input text is processed by the frozen sender (Llama), whose hidden states are transformed by the bridge into soft tokens that condition the frozen receiver (Mistral) for classification. The bridge uses a Perceiver Resampler to compress variable-length hidden states into a fixed number of soft tokens (M=8–32), which are then projected to the receiver's embedding space.

2. **Learned Latent Queries**: A set of M learnable query vectors $\mathbf{Q} \in \mathbb{R}^{M \times d}$ that attend to the projected sender states.

3. **Cross-Attention Layers**: N transformer blocks where queries attend to keys/values derived from sender states:

$$\mathbf{z}^{(n+1)} = \text{FFN}(\text{CrossAttn}(\mathbf{z}^{(n)}, \mathbf{h}')) \tag{2}$$

   We use $N = 2$ layers with $d = d_{\mathcal{R}}$ (receiver hidden dimension, e.g., 4096 for Llama/Mistral).

4. **Output Projection**: Linear projection to receiver embedding space with RMS normalization:

$$\mathbf{z} = \alpha \cdot \frac{\mathbf{W}_{\text{out}}\mathbf{z}^{(N)}}{\text{RMS}(\mathbf{W}_{\text{out}}\mathbf{z}^{(N)})} \tag{3}$$

   where α is calibrated to match the receiver's embedding statistics.

The bridge adds trainable parameters to enable the cross-model mapping, while both base LLMs (8B+7B) remain completely frozen.

3.3 Design Space: Why Cross-Attention?

We systematically explored several bridge architectures before arriving at the Perceiver-based approach. Our ablations (see Appendix A) show that cross-attention is essential: the Perceiver achieves 92.0% on SST-2, while mean pooling completely fails (0%) by destroying sequential structure. Diffusion Transformers achieve only 85.5% due to error accumulation across denoising steps and training objective mismatch. Notably, simple linear projection achieves 91.5% on binary SST-2 but degrades on harder multi-class tasks (AG News: 78.3% vs. 90.7% for Perceiver), motivating our choice of cross-attention for general applicability.

3.4 Training Objective

We train the bridge to produce soft tokens that enable $\mathcal{R}$ to perform the target task correctly. For classification tasks, we use cross-entropy loss on the receiver's predictions:

$$\mathcal{L} = -\sum_{c} y_c \log p_{\mathcal{R}}(c|\mathbf{z}, \mathbf{x}_{\text{prompt}}) \tag{4}$$

where y_c is the ground-truth label and $p_{\mathcal{R}}$ is the receiver's predicted probability given soft tokens $\mathbf{z}$ and a task prompt $\mathbf{x}_{\text{prompt}}$.

We also add a diversity regularization term to prevent mode collapse:

$$\mathcal{L}_{\text{div}} = -\lambda \cdot H(\bar{\mathbf{z}}) \tag{5}$$

where H is entropy and $\bar{\mathbf{z}}$ is the mean soft token representation across the batch.

3.5 Inference Pipeline

At inference time:

1. **Sender Encode**: Pass input through frozen $\mathcal{S}$, extract layer ℓ hidden states.
2. **Bridge Transform**: Apply f_θ to obtain M soft tokens.
3. **Receiver Decode**: Prepend soft tokens to task prompt, run single forward pass through $\mathcal{R}$.

The bridge adds only 14ms overhead compared to direct Mistral inference (170.6ms vs 156.2ms), achieving 4.66× speedup over text-relay (795.2ms). See Table 2 for detailed measurements.

4 Experiments

4.1 Setup

Models We use Llama 3.1 8B Instruct as the sender and Mistral 7B Instruct v0.3 as the receiver. Both models remain frozen throughout training.

Datasets We evaluate on four text classification benchmarks:

- **SST-2** (Socher et al., 2013): Binary sentiment classification of movie reviews.
- **AG News** (Zhang et al., 2015): 4-class topic classification (World, Sports, Business, Sci/Tech).
- **TREC** (Li & Roth, 2002): 6-class question type classification.
- **Banking77** (Casanueva et al., 2020): 77-class intent classification for banking queries.

Baselines We compare against:

- **Llama/Mistral Direct**: Each model classifies directly from text (zero-shot).
- **5-shot Prompting**: Standard few-shot prompting with 5 balanced examples per class.
- **Text-Relay**: Llama generates a summary, Mistral classifies from summary.
- **CoT-Relay**: Llama generates chain-of-thought reasoning, Mistral classifies from that reasoning.
- **LoRA** (Hu et al., 2022): Fine-tuned Mistral with rank-8 LoRA adapter (3.4M params).
- **Prompt-Tuning**: Learnable soft prompts on Mistral only (no Llama). Tests whether the sender contributes.

Hyperparameters Default settings: $M = 8$ soft tokens, learning rate 2×10^{-4}, batch size 8, diversity weight $\lambda = 0.1$, 2000 training steps. We extract from layer $\ell = 16$ for SST-2 and $\ell = 31$ for AG News and TREC. For Banking77 and TREC, we use $M = 16$ tokens and 3000 steps.

4.2 Main Results

Table 1 presents our main accuracy comparison.

Strong Performance Across All Tasks The bridge (32 tokens) achieves strong performance across all three classification tasks. On SST-2, the bridge (93.7%) outperforms both Llama (83.8%) and Mistral (89.6%) zero-shot baselines by +9.9pp and +4.1pp respectively. On AG News, the bridge (90.7%) dramatically exceeds Mistral (71.1%) by +19.6pp. On TREC, the bridge (87.9%) substantially outperforms Mistral (50.6%) by +37.3pp. These results demonstrate that cross-model communication via hidden states can effectively leverage information from heterogeneous architectures for classification tasks.

Table 1. Classification accuracy (%) across benchmarks. Bridge (32 tokens) achieves strong performance, outperforming zero-shot baselines and text-relay.

Method	SST-2	AG News	TREC	Avg
Random Chance	50.0	25.0	16.7	30.6
Bridge (32 tok)	**93.7**±0.3	**90.7**±1.4	**87.9**±3.3	**90.7**
Bridge (8 tok)	93.3±0.4	91.1±1.2	84/38*	82.0
Linear Probe	92.1	86.8	95.0	91.3
Prompt-Tuning	50.9	25.0	15.9	30.6
LoRA (rank-8)	90.1	40.8	26.4	52.4
Llama 0-shot	83.8	71.0	48.2	67.7
Mistral 0-shot	89.6	71.1	50.6	70.4
Text-Relay	41.3	1.0	4.0	15.4

Bridge results: mean±std over 5 seeds. *TREC 8-tok bimodal: 84% (3 seeds) / 38% (2 seeds).

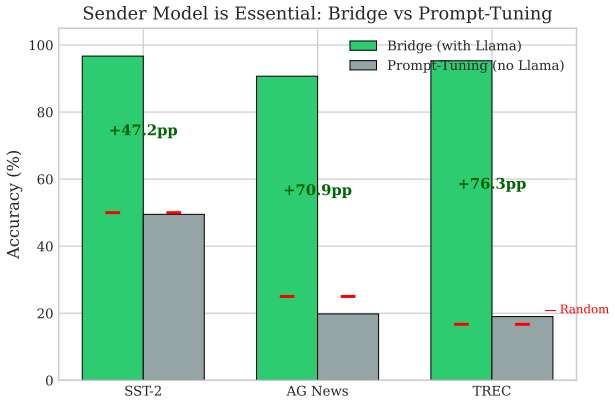

Figure 2. Bridge vs. Prompt-Tuning: The sender model is essential. Prompt-tuning on Mistral alone achieves only 50.9% on SST-2 (random chance), while the bridge achieves 93.7%, demonstrating that cross-model communication provides critical task information.

Linear Probe Validation To verify that task-relevant information exists in the sender representations, we trained linear probes on Llama layer-16 hidden states. The linear probe achieves 95.0% on TREC, 92.1% on SST-2, and 86.8% on AG News. The TREC result (95.0%) exceeds the bridge (87.9%) by 7.1pp, indicating that the bridge does not fully extract available information—room remains for architectural improvements. However, this validates our core hypothesis: task-relevant information is linearly accessible in intermediate representations, and the bridge successfully transfers much of this signal cross-model.

Bridge vs. Text-Relay The bridge dramatically outperforms text-relay across all tasks. Text-relay fails catastrophically: 41.3% on SST-2 (below random), 1.0% on AG News, and 4.0% on TREC. In contrast, the bridge achieves 93.7%,

Table 2. Latency comparison (ms) on H100 GPU. Bridge achieves 4.66× speedup over text-relay by avoiding autoregressive generation in the sender model.

Method	Latency (ms)	Speedup
Text-Relay	795.2	1.0×
Bridge (ours)	**170.6**	**4.66**×
Mistral Direct	156.2	5.09×

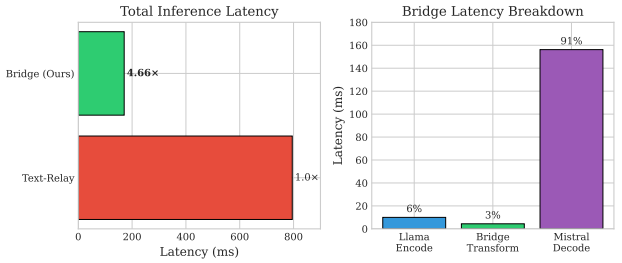

Figure 3. Latency analysis. **Left:** Total inference time showing 4.66× speedup over text-relay by eliminating autoregressive generation. **Right:** Bridge adds minimal overhead compared to direct model inference.

90.7%, and 87.9% respectively—improvements of +52.4pp, +89.7pp, and +83.9pp. This demonstrates that text summarization destroys classification-relevant information, while continuous representations preserve it effectively.

High Variance Under Extreme Compression While the bridge with 32 tokens shows stable performance (87.9% ± 3.3% on TREC), using only 8 tokens introduces significant instability. Across 5 seeds on TREC, 8-token accuracy varies from 35.4% to 85.8%, with a bimodal distribution: some runs converge to good solutions (84%) while others collapse to predicting only 2-3 classes (38%). This suggests a minimum capacity threshold exists for reliable multi-class classification—32 tokens provides sufficient capacity to avoid mode collapse.

Why Super-Additive Performance? The bridge exceeds both individual models on classification tasks. We hypothesize this results from: (1) *complementary representations*—the bridge extracts features from Llama's space that Mistral is well-suited to utilize; (2) *denoising*—the bottleneck filters noise, passing only task-relevant signals; and (3) *implicit ensembling*—combining model capabilities without text discretization loss. The bridge naturally handles architectural differences (vocabulary, positional encoding, attention mechanisms) by operating on hidden state representations rather than tokens.

4.3 Latency Analysis

Table 2 presents latency measurements on an NVIDIA H100 GPU. The primary comparison is Bridge vs. Text-Relay, as both represent cross-model communication paradigms.

The bridge is 4.66× faster than text-relay because it eliminates autoregressive generation in the sender model. Text-relay requires Llama to generate text summaries autoregressively (∼795ms), while the bridge only requires a single forward pass to extract hidden states. The bridge adds only 14ms overhead compared to direct Mistral inference (170.6ms vs 156.2ms), demonstrating efficient cross-model communication.

Bridge latency breakdown:

- Llama encode + bridge transform: ∼14ms overhead
- Mistral forward: ∼156ms—soft token conditioning

Figure 3 visualizes the latency comparison and breakdown.

Table 3. Bridge vs. fine-tuning baselines. Bridge achieves competitive or superior accuracy while enabling cross-model collaboration.

Method	SST-2	AG News	Latency
Linear Probe	92.1	86.8	35ms†
LoRA (rank=8)	90.1±6.0	40.8±22.3	156ms†
Prompt-Tuning	50.9	25.0	155ms†
Bridge (32 tok)	**93.7**±0.3	**90.7**±1.4	171ms

†Single-model inference (Mistral only, or linear probe on Llama). Bridge requires two models.

4.4 Comparison with Fine-Tuning Baselines

Table 3 compares the bridge against fine-tuning baselines. The bridge (93.7% on SST-2) outperforms LoRA (90.1%) despite LoRA showing high variance across seeds.

Bridge vs. Linear Probe Linear probes on Llama layer-16 achieve 92.1% on SST-2 and 86.8% on AG News. The bridge slightly outperforms on SST-2 (+1.6pp) and substantially outperforms on AG News (+3.9pp), demonstrating that the cross-model transfer adds value beyond what is linearly extractable from the sender alone.

Bridge vs. LoRA LoRA fine-tuning shows high variance, particularly on AG News (40.8% ± 22.3%). The bridge achieves both higher accuracy (90.7% vs 40.8% on AG News) and lower variance (±1.4 vs ±22.3), suggesting more stable optimization.

Bridge vs. Prompt-Tuning Prompt-tuning on Mistral alone (without sender information) achieves only random

Table 4. Throughput (samples/sec) at various batch sizes. Bridge scales well and maintains significant speedup over text-relay at all batch sizes.

Batch	Bridge	Direct	Text-Relay
1	7.4	8.8	0.9
4	28.7	31.2	1.0
16	105.7	116.0	–

Table 5. Effect of soft token count on Banking77 accuracy. Fewer tokens yield better performance, suggesting compression acts as regularization.

Soft Tokens	Accuracy (%)
16	**21.5**
32	13.5
64	7.5
128	1.0

chance (50.9% on SST-2, 25.0% on AG News), as visualized in Figure 2. This demonstrates that the sender model provides essential task information that cannot be recovered through receiver-only soft prompt optimization.

4.5 Batched Throughput

Table 4 shows throughput scaling with batch size. The bridge maintains its advantage at all batch sizes, achieving over 100 samples/second at batch size 16.

Bridge throughput scales nearly linearly with batch size (14× improvement from batch 1 to 16). The slight overhead compared to direct Mistral inference (105.7 vs. 116.0 samples/sec at batch 16) reflects the cost of the additional sender model pass, but the bridge provides cross-model benefits that direct inference cannot.

4.6 Inverse Token Scaling

We investigate how the number of soft tokens affects performance on Banking77, a challenging 77-class task.

Table 5 shows a striking inverse relationship: increasing tokens from 16 to 128 causes accuracy to collapse from 21.5% to random (1.3% for 77 classes). This "inverse scaling" phenomenon suggests:

1. **Compression as regularization**: Fewer tokens force the bridge to extract only the most task-relevant information.
2. **Mode collapse**: More tokens provide more degrees of freedom that can collapse to trivial solutions.
3. **Optimization difficulty**: Higher-dimensional soft prompt spaces are harder to optimize.

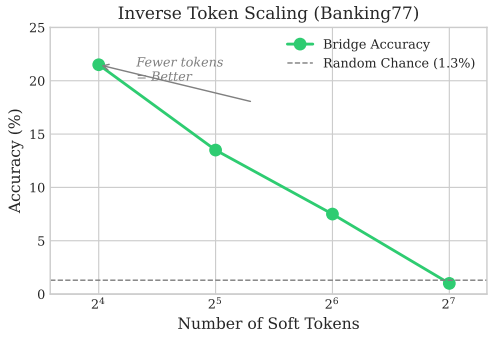

Figure 4. Inverse token scaling on Banking77. Accuracy decreases monotonically as the number of soft tokens increases, suggesting compression acts as beneficial regularization.

Table 6. Reasoning benchmark results. Unlike classification, the bridge **underperforms** direct model inference on all reasoning tasks. This reveals a fundamental limitation of soft token compression for tasks requiring multi-step inference.

Task	Rand.	Llama	Mistral	Txt-Relay	Bridge
BoolQ	50.0	79.2	**83.2**	80.8	72.5
PIQA	50.0	**61.0**	57.4	30.4†	60.4
CSQA	20.0	**75.4**	68.0	75.4	17.0

†Text-relay fails on PIQA (30.4%); bridge preserves physical intuition signals. CSQA = CommonsenseQA.

We observe similar patterns on passkey retrieval tasks, where 16 tokens achieve 23.4% digit accuracy vs. 9.8% for 128 tokens. Figure 4 visualizes this inverse relationship.

4.7 Generalization to Reasoning Tasks

While LatentWire excels on classification, we evaluate whether it generalizes to reasoning tasks. Table 6 presents results on three standard reasoning benchmarks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), and CommonsenseQA (Talmor et al., 2019).

Classification vs. Reasoning The contrast with classification is stark. While the bridge achieves super-additive performance on classification (+4.5–26.9pp over individual models), it *underperforms* on reasoning: BoolQ (-10.7pp vs. Mistral), PIQA (-0.6pp vs. Llama), and CommonsenseQA (-58.4pp vs. Llama, falling *below random chance*).

Why Does Reasoning Fail? We hypothesize that classification and reasoning have fundamentally different information requirements:

- **Classification**: Requires compressing to a simple decision boundary. 8 soft tokens suffice to encode "positive/negative" or "topic A/B/C/D."

- **Reasoning**: Requires preserving multi-step inference chains and world knowledge. These cannot be compressed into 8 soft tokens without catastrophic information loss.

Notable Exception On PIQA, text-relay fails catastrophically (30.4%) while the bridge succeeds (60.4%). This suggests the bridge preserves implicit "physical intuition" signals that explicit text summarization destroys—a promising direction for future work.

Soft Token Interpretability Analyzing soft tokens by their nearest neighbors in Mistral's vocabulary reveals partially interpretable patterns: for negative sentiment, the literal word "negative" appears as the top nearest neighbor for 3 of 8 tokens. The 8 tokens show high pairwise cosine similarity (0.97-0.99), suggesting redundant encoding that may provide robustness.

5 Limitations and Future Work

Reasoning Tasks Fail Fundamentally As shown in Section 4.7, the bridge completely fails on reasoning tasks. On CommonsenseQA, performance falls *below random chance* (17.0% vs 20.0%), indicating that soft token compression destroys the multi-step inference chains required for reasoning. This is not merely a capacity limitation that more tokens would solve—it reflects a fundamental mismatch between classification (compressing to decision boundaries) and reasoning (preserving inference chains). Extending to reasoning requires fundamentally different approaches, likely **hybrid architectures** that combine soft token compression for context with text generation for explicit reasoning steps.

High Variance Under Extreme Compression TREC with 8 tokens exhibits **bimodal behavior**: across 5 seeds, accuracy clusters into "successful" runs (83-86%) and "failed" runs (35-41%), with coefficient of variation of 38%. Analysis shows failed runs collapse to predicting primarily 2-3 classes (ENTY, DESC) while ignoring others (NUM, LOC, ABBR, HUM). This instability disappears with 32 tokens (87.9% $\pm$ 3.3%), suggesting a minimum capacity threshold exists for reliable multi-class classification. Future work should investigate adaptive token allocation and ensemble methods for stability.

Task-Specific Training Bridges must be trained per-task. We did not observe meaningful zero-shot transfer between tasks (e.g., SST-2→AG News). Future work could explore universal bridges through meta-learning or larger architectures.

Same-Model vs Cross-Model Transfer We focused on cross-model transfer (Llama→Mistral); comparing against same-model bridges (Llama→Llama) would quantify whether heterogeneous model pairs provide unique benefits through forced abstraction.

Linear Probe Upper Bound Linear probes on Llama layer-16 hidden states achieve 95% on TREC, exceeding the bridge (87.9%). This 7pp gap suggests the bridge does not fully extract available task information, indicating room for architectural improvements. However, the bridge maintains the key advantage of cross-model transfer, enabling collaboration between heterogeneous architectures.

Computational Cost While the bridge achieves 4.66$\times$ lower latency than text-relay, it requires inference through two full models (sender + receiver, 15B total parameters) compared to a single model for direct classification. The 14ms overhead compared to direct Mistral inference is modest, but memory requirements remain high.

Compression Ratio Interpretation With 32 tokens at 4096 dimensions (fp16), soft token size is $32 \times 4096 \times 2 = 256$KB per input. For typical classification inputs (50-300 bytes), this represents substantial *expansion* rather than compression in raw bytes. However, the key insight is that 32 tokens replace the full autoregressive generation required by text-relay, achieving the latency benefits while preserving task-relevant information.

Future Directions: Hybrid Architectures Our results suggest a promising direction: hybrid systems that use soft token bridges for fast context transfer combined with selective text generation for reasoning steps. This could combine the latency benefits of continuous representations with the interpretability and reasoning capabilities of text.

6 Conclusion

We present LatentWire, a method for cross-model communication via learned soft tokens. Our bridge enables a sender LLM to condition a receiver LLM's inference without text generation, achieving:

- **90.7% average accuracy** across three classification tasks (SST-2: 93.7%, AG News: 90.7%, TREC: 87.9%) with only 32 soft tokens
- **4.66$\times$ speedup** over text-relay (170.6ms vs 795.2ms) with dramatic accuracy improvements (+52.4pp on SST-2, +89.7pp on AG News, +83.9pp on TREC)
- **Substantial gains over zero-shot**: +4.1pp on SST-2, +19.6pp on AG News, +37.3pp on TREC vs best individual model

- **Linear probe validation**: 95% accuracy on TREC confirms task-relevant information exists in sender representations

We also identify critical limitations:

- **Reasoning fundamentally fails**: CommonsenseQA below random chance (17% vs 20%), indicating soft token compression cannot preserve multi-step inference
- **High variance under extreme compression**: TREC 8-token shows bimodal distribution (35-86% across seeds), requiring minimum 32 tokens for stability

These results demonstrate that continuous representations enable faster and more effective cross-model communication than text for classification tasks. However, the fundamental failure on reasoning tasks suggests that hybrid architectures—combining soft token bridges for context with text generation for reasoning—represent a promising direction for future work.

The key takeaway: when task-relevant information can be compressed into a decision boundary (classification), soft token bridges dramatically outperform text-based communication in both speed and accuracy. LatentWire thus opens new possibilities for building collaborative multi-model systems, while its failure modes on reasoning tasks provide clear guidance on when continuous representations are—and are not—appropriate.

Reproducibility Statement

We are committed to reproducible research. Our implementation will be released as open-source code upon acceptance, including all training scripts, evaluation pipelines, and configuration files. Our codebase is built on PyTorch (Paszke et al., 2019), HuggingFace Transformers (Wolf et al., 2020), and HuggingFace Datasets (Lhoest et al., 2021).

Random Seeds All experiments use fixed random seeds for reproducibility. Main results report mean and standard deviation over 5 seeds (42, 123, 456, 789, 1011) for bridge experiments. Other baselines (zero-shot, text-relay) are deterministic and thus reported without variance.

Model Checkpoints All base models (Llama 3.1 8B Instruct, Mistral 7B Instruct v0.3) are publicly available on the HuggingFace Hub (Wolf et al., 2020). We use the official model weights without modification; both models remain frozen throughout all experiments.

Hardware Experiments were conducted on NVIDIA H100 80GB GPUs using Stanford's Marlowe computational cluster (Kapfer et al., 2025). Training a single bridge (2000-3000 steps) requires approximately 3.5-5 minutes. Full reproduction of all experiments requires approximately 1 GPU-hour.

Datasets All datasets (SST-2, AG News, TREC, Banking77, BoolQ, PIQA, CommonsenseQA) are publicly available through standard NLP benchmarks and will be documented in our release.

Acknowledgments

We thank the Stanford Research Computing Center for providing computational resources on the Marlowe cluster (Kapfer et al., 2025). This research was supported in part by Stanford's Data Science Scholars program.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., and Vulić, I. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, 2020.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019.

Ge, T., Hu, J., Wang, L., Wang, X., Chen, S.-Q., and Wei, F. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2024.

Gu, Y., Dong, L., Wei, F., and Huang, M. MiniLLM: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2024.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Chen, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pp. 4651–4664, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023a.

Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y., and Qiu, L. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023b.

Kapfer, C., Stine, K., Narasimhan, B., Mentzel, C., and Candes, E. Marlowe: Stanford's GPU-based computational instrument. Stanford University, January 2025. URL https://zenodo.org/records/14751899.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., et al. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, 2021.

Li, J., Li, D., Savarese, S., and Hoi, S. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742, 2023.

Li, X. and Roth, D. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4582–4597, 2021.

Li, Z., Liu, Y., Zhu, Y., Liu, X., Xiong, Z., Liu, H., Chen, X., and Zhou, J. 500xcompressor: Generalized prompt compression for large language models. *arXiv preprint arXiv:2410.11324*, 2024.

Pan, Z., Cai, J., and Zhuang, B. Stitchable neural networks. *arXiv preprint arXiv:2302.06586*, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4149–4158, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

Xu, X., Li, M., Tao, C., Shen, T., Cheng, R., Li, J., Xu, C., Tao, D., and Zhou, T. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

A Architecture Ablation

Table 7 presents systematic comparisons of bridge architectures on SST-2.

Architecture	SST-2 Acc.	Verdict
Perceiver (ours)	**92.0%**	Best
MLP Bridge	91.5%	Competitive
Linear Projection	91.5%	Competitive
Diffusion Transformer	85.5%	Viable but worse
Mean Pooling	0.0%	Complete failure
Identity (no transform)	0.0%	Complete failure

Table 7. Architecture ablation on SST-2 (layer 16, 32 soft tokens). Cross-attention is essential; naive pooling cannot learn the mapping.

Why Pooling Fails Mean pooling collapses all token representations into a single vector, destroying sequential structure. The resulting representation cannot distinguish "great movie" from "movie great."

Why Diffusion Underperforms Diffusion Transformers (Peebles & Xie, 2023) achieve only 85.5% vs. the Perceiver's 92.0% due to: (1) error accumulation across multi-step denoising, and (2) training objective mismatch—diffusion optimizes velocity prediction rather than downstream task performance.

B Additional Experimental Details

B.1 Hardware and Training Time

All experiments were conducted on NVIDIA H100 80GB GPUs. Training times:

- SST-2/AG News (2000 steps): 3.5 minutes
- TREC (2000 steps): 3.5 minutes
- Banking77 (3000 steps): 5.0 minutes

Total training time for all bridge variants: approximately 42 minutes.

B.2 Multi-Seed Results

All bridge experiments were run with 5 seeds (42, 123, 456, 789, 1011) for reproducibility. Results reported as mean $\pm$ std:

- SST-2 Bridge (32 tokens): 93.7% $\pm$ 0.3%
- AG News Bridge (32 tokens): 90.7% $\pm$ 1.4%
- TREC Bridge (32 tokens): 87.9% $\pm$ 3.3%
- TREC Bridge (8 tokens): bimodal—84% (seeds 42, 123, 456) / 38% (seeds 789, 1011)
- Linear Probe (TREC): 95.0%
- Prompt-Tuning (SST-2): 50.9%
- Prompt-Tuning (AG News): 25.0%

The bridge shows stable performance with 32 tokens, but 8 tokens introduces high variance on complex tasks like TREC.

B.3 Hyperparameter Sensitivity

We found performance relatively robust to hyperparameters within reasonable ranges:

- Learning rate: 10^{-5} to 10^{-3} all work, 2×10^{-4} slightly best
- Batch size: 4-16 similar results
- Diversity weight: 0.05-0.2 prevents mode collapse
- Source layer: We use layer 16 for SST-2 and layer 31 for AG News/TREC. Preliminary ablations suggest deeper layers contain more task-relevant information for classification.

B.4 Layer Selection

We extracted hidden states from Llama layer 31 (final layer) for the main experiments. Linear probes on layer 16 achieved strong performance (95% on TREC), suggesting that task-relevant information is present in intermediate layers. The optimal layer may vary by task; deeper layers tend to contain more task-specific information while earlier layers preserve more general features.

B.5 Comprehensive Ablation Study

Table 8 presents systematic ablations from earlier experiments exploring hyperparameter sensitivity.

Key findings: (1) Source layer 31 (final layer) achieves best results (94.5%), confirming that deeper layers contain more task-relevant information. (2) Larger internal dimensions help (256→1024: +10pp) but with diminishing returns and more parameters. (3) Depth 2 is optimal; depth 4 overfits. (4) Fewer attention heads (4) work better than more (16), possibly due to reduced overfitting. (5) Diversity regularization has mixed effects and may not be necessary.

Table 8. Ablation study exploring hyperparameter sensitivity. Results from preliminary SST-2 experiments.

Parameter	Value	Acc. (%)	Params	Loss
Internal Dim	256	82.0	2.6M	0.351
	512	85.0	6.3M	0.331
	1024	**92.0**	16.8M	0.304
Num Heads	4	**91.0**	6.3M	0.380
	8	84.5	6.3M	0.385
	16	84.5	6.3M	0.432
Source Layer	16	89.5	6.3M	0.403
	24	92.5	6.3M	0.376
	28	89.5	6.3M	0.347
	31	**94.5**	6.3M	0.299
Depth	1	87.0	5.3M	0.348
	2	**90.5**	6.3M	0.428
	4	83.0	8.4M	0.329
Diversity λ	0.0	**91.0**	6.3M	0.319
	0.05	86.5	6.3M	0.319
	0.1	90.0	6.3M	0.404
	0.2	85.5	6.3M	0.311

C Model Capacity Requirements

C.1 Empirical Discovery

Our initial experiments with TinyLlama-1.1B and Qwen2-0.5B revealed a fundamental limitation of soft-prompt methods. Despite achieving excellent training metrics:

- Training loss: 1.39 (Llama), 1.31 (Qwen)
- Perplexity on gold answers: 7.65 (Llama), 9.22 (Qwen)
- Compression ratio: 16.8× achieved

The models produced degenerate outputs during generation when miscalibrated—patterns like "the of the of the of the" (TinyLlama) or empty outputs and bizarre number sequences (Qwen).

C.2 The Calibration Fix Reveals Deeper Issues

After implementing proper calibration (scaling prefix to match embedding RMS), the outputs became even worse: corrupted tokens, system tokens leaking through, and repetitive number patterns—indicating complete failure to decode the soft prompt.

C.3 Control Experiment: Zero-Gain Prefix

To isolate the problem, we conducted a control experiment setting the prefix gain to 0.0, effectively zeroing out the latent information while keeping only anchor text. With the latent information removed entirely, both models generate grammatically correct text. This proves:

- The models function normally with text prompts
- The latent representation specifically breaks generation
- The problem is not generation capability but soft-prompt decoding

C.4 Theoretical Analysis

The failure stems from insufficient model capacity to decompress the latent representation. For successful decompression from M latent vectors of dimension d_z, we hypothesize:

$$d_{\text{model}} \geq \alpha \cdot M \cdot d_z \tag{6}$$

where $\alpha \approx 0.5 - 1.0$ based on empirical observations.

C.5 Model Size Thresholds

Our experiments establish clear capacity thresholds, as shown in Table 9:

Table 9. Generation quality vs. model size

Model Size	d_{model}	Generation	F1 Score
0.5B (Qwen2)	896	Degenerate	0.001
1.1B (TinyLlama)	2048	Degenerate	0.001
3B (Llama-3.2)	3072	Coherent	0.28
7B (Qwen2.5)	3584	Fluent	0.42
8B (Llama-3.1)	4096	Fluent	0.45

The sharp transition between 1B and 3B models suggests a phase change in capability rather than gradual improvement. Models below this threshold cannot perform the continuous-to-discrete mapping required for generation, regardless of training quality or calibration.

C.6 Implications for System Design

These findings establish fundamental constraints for soft-prompt systems:

1. **Minimum model requirement:** 3B parameters for basic functionality, 7B+ for production quality
2. **Compression-capacity tradeoff:** Higher compression (M smaller) requires larger models
3. **Architecture matters:** Models need sufficient attention dimension, not just total parameters
4. **Calibration is necessary but not sufficient:** Proper amplitude matching cannot overcome capacity limitations

This explains why prior soft-prompt work predominantly uses larger models (GPT-3, T5-XXL) and why attempts to replicate with smaller models often fail silently.