HMU: Hippocampus Memory Unit
A Semantic Memory Module for Transformers

Esteban Sánchez Gámez

estebansanchezgamez@gmail.com
https://github.com/Sauvageduck24

Abstract. The *Hippocampus Memory Unit* (HMU) is a bio-inspired module that expands the capabilities of Transformer models by introducing a compact, trainable semantic latent space. Taking the function of the human hippocampus—responsible for consolidating and evoking memories—as a reference, HMU acts as an adaptive semantic memory capable of: (i) condensing relevant information, (ii) filtering contextual noise, and (iii) enriching creative generation. With an approximate parameter overhead of 1 %, the resulting model increases long-term memory retention, narrative coherence, and accuracy in classification and generation tasks, *without the need to retrain* the encoder or decoder. A slight *fine-tuning* of the HMU block and its associated VAE is sufficient.

1 Introduction

Transformer models have set a new standard in natural language processing (NLP) tasks [1]. However, despite their success, they present persistent limitations in three key areas: (i) difficulty maintaining context in long sequences, (ii) limited diversity and creativity in generation, and (iii) absence of explicit semantic memory or contextual compression mechanisms.

These shortcomings limit their performance in tasks that require long-term reasoning, narrative generation, or efficient handling of redundant information. Inspired by neuroscience—where the hippocampus does not store long-term memory directly, but rather decides *when* and *which* memories to reactivate based on context [4]—we propose HMU, a lightweight, plug-and-play module that introduces a form of controlled semantic latent memory.

This module, based on a token-level VAE and a trainable adaptive gate, can be inserted without modifying the encoder or decoder, allowing existing models to be extended with memory, compression, and imagination capabilities without the need for structural retraining.

Contributions This work presents:

1. the design of the HMU module, inspired by the selective role of the hippocampus,
2. its minimal and non-invasive integration into Transformer architectures,
3. an empirical evaluation in classification tasks, narrative generation, and synthetic memory benchmarks, and
4. a cost-benefit analysis demonstrating that a mere +1 % increase in parameters can translate into improvements of up to 20x in long-term retention.

2 HMU Architecture

2.1 General Description

HMU Module (Hippocampus Memory Unit). The HMU sits as a *semantic filter* between the encoder and the decoder: it receives the contextual output from the encoder, $\mathbf{v} \in \mathbb{R}^{B \times L \times d_{\text{model}}}$, and delivers a version to the decoder enriched $\mathbf{v}^*$.

1. Latent compression with VAE. For each token a *Variational Autoencoder* is calculated:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) = \text{VAE-enc}\big(\mathbf{v}\big), \qquad \mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

and the compressed version is rebuilt $\tilde{\mathbf{v}} = \text{VAE-dec}(\mathbf{z})$.

2. Normalization. Both paths are normalized to avoid scale imbalances:

$$\hat{\mathbf{v}} = \text{LayerNorm}(\mathbf{v}), \qquad \hat{\mathbf{v}}_{\text{lat}} = \text{LayerNorm}(\tilde{\mathbf{v}}).$$

3. Adaptive Gating. The literal and latent information is concatenated, and a *MLP sigmoid* produces the gate

$$\mathbf{g} = \sigma\Big(\text{MLP}\big[\hat{\mathbf{v}} \,\|\, \hat{\mathbf{v}}_{\text{lat}}\big]\Big), \qquad \mathbf{g} \in [0,1]^{B \times L \times d_{\text{model}}}.$$

4. Semantic fusion. The final representation is calculated on an element-by-element basis:

$$\mathbf{v}^* = \mathbf{g} \odot \hat{\mathbf{v}} + (1 - \mathbf{g}) \odot \hat{\mathbf{v}}_{\text{lat}},$$

followed by a linear projection, activation GELU and LayerNorm to maintain the dimensionality d_{model}. This stage constitutes the final output of the HMU module and ensures that the representation is normalized and ready to be consumed by the *decoder*.

Resulting properties.

- *Controlled Imaging*: The VAE's continuous latent space enables the generation of coherent semantic variations, tunable by parameters such as temperature or sampling type.
- *Selective Memory*: The learned gating mechanism simulates the function of the hippocampus, contextually and dynamically deciding *how much* compressed information should be retrieved and *when* it should be integrated.
- *Efficient Integration*: The HMU module introduces an overhead of just $\sim 1\,\%$ to the total number of parameters, and can be coupled to existing Transformer models without altering their internal structure.

Figure 1 illustrates the complete flow.

2.2 Reference implementation with Pytorch

```
class HMU(nn.Module):
    def __init__(self, d_model: int, latent_dim: int):
        super().__init__()
        self.vae = VAE(d_model, latent_dim)

        # Normaliza el input original y el latente reconstruido
        self.norm_input = nn.LayerNorm(d_model)
        self.norm_latent = nn.LayerNorm(d_model)

        # Gating para decidir cu nto usar de cada representaci n
        self.gate_mlp = nn.Sequential(
            nn.Linear(d_model * 2, d_model),
            nn.Sigmoid()
        )

        # Proyecci n final tras fusi n
        self.fusion_proj = nn.Sequential(
            nn.Linear(d_model, d_model),
            nn.GELU(),
```

```
            nn.LayerNorm(d_model)  # Prepara para decoder
        )

    def forward(self, v: torch.Tensor) -> torch.Tensor:
        """
        Args:
            v (Tensor): Salida del encoder (batch, seq_len, d_model)
        Returns:
            Tensor: Representaci n enriquecida para el decoder
        """
        # VAE reconstruction
        v_latent = self.vae(v)  # (batch, seq_len, d_model)

        # Normalizar ambas representaciones
        v_norm = self.norm_input(v)
        v_latent_norm = self.norm_latent(v_latent)

        # Gating: decide cu nto confiar en cada parte
        gate_input = torch.cat([v_norm, v_latent_norm], dim=-1)  # (batch, seq_len, 2*d_model)
        gate = self.gate_mlp(gate_input)  # (batch, seq_len, d_model), valores entre 0 y 1

        # Fusi n adaptativa
        fused = gate * v_norm + (1 - gate) * v_latent_norm

        # Proyecci n final
        output = self.fusion_proj(fused)  # (batch, seq_len, d_model)
        return output
```

2.3 Mathematical flow of the HMU module

The HMU module takes as input a sequence of representations $\mathbf{h} \in \mathbb{R}^{B \times L \times d_{\text{model}}}$, and generates an enriched version $\mathbf{h}^*$ ready for use in the decoder. The process consists of the following steps:

$$\tilde{\mathbf{h}} = \text{VAE}(\mathbf{h}) \qquad \text{(Latent compression)} \tag{1}$$

$$\hat{\mathbf{h}} = \text{LayerNorm}(\mathbf{h}) \qquad \text{(Original Normalization)} \tag{2}$$

$$\hat{\tilde{\mathbf{h}}} = \text{LayerNorm}(\tilde{\mathbf{h}}) \qquad \text{(Latent Normalization)} \tag{3}$$

$$\mathbf{g} = \sigma\left(W_g \left[\hat{\mathbf{h}}\,;\, \hat{\tilde{\mathbf{h}}}\right]\right) \qquad \text{(Adaptive gate)} \tag{4}$$

$$\mathbf{h}' = \mathbf{g} \odot \hat{\mathbf{h}} + (1 - \mathbf{g}) \odot \hat{\tilde{\mathbf{h}}} \qquad \text{(Semantic fusion)} \tag{5}$$

$$\mathbf{h}^* = \text{LayerNorm}\left(\phi\left(W_f \mathbf{h}'\right)\right) \qquad \text{(Final Projection)} \tag{6}$$

where:

[topsep=2pt,itemsep=2pt,leftmargin=14pt]VAE: Token-level variational autoencoder. $[\,;\,]$: Concatenation along the feature dimension. σ: Sigmoid function applied element by element. ϕ: GELU activation function. $W_g \in \mathbb{R}^{2d \times d}, W_f \in \mathbb{R}^{d \times d}$: Learned projections.

3 Experimental Methodology

HMU was evaluated on five datasets: GLUE, AG-News, CommonGen, ROCStories, and a synthetic memory benchmark. Hyperparameters were kept identical to the baseline except for the inclusion of the HMU (see Table 1).

Table 1: Datasets used.

Category	Dataset	Purpose
Classification	GLUE, AG-News	Comprehension and Classification
Generation	CommonGen	Concept-Based Generation
Narrative	ROCStories	Narrative Coherence
Memory	Synthetic	Retention@200

4 Results

This section presents a quantitative comparison between the standard Transformer and its variant with the integrated HMU unit. The evaluation covers classification tasks, text generation, and synthetic benchmarks focused on long-term memory. The models have been trained with the same hyperparameters and configurations to ensure a fair comparison, with only the internal architecture being modified by the inclusion of the HMU module.

4.1 Quantitative Comparison by Task

GLUE (Reduced Size). On a truncated subset of GLUE ($\leq$ 20,000 examples per task), the model with HMU achieves an average improvement of **+0.6 percentage points** in *accuracy*. Significant improvements are noted in syntactic tasks such as CoLA (+7.5 pp) and paraphrasing (MRPC, +2.0 pp). Some tasks with a heavy semantic load, such as SST-2 and MNLI, show slight losses attributable to overfitting or lack of convergence.

Synthetic Memory Benchmark. Long-term retention is assessed using the RETENTION@200 metric, based on the cosine similarity between the input and output after 200 propagation steps. The incorporation of HMU provides a **20-fold** improvement in this metric, demonstrating much more effective retrieval of the original context. An improvement in *sequential coherence* and retrieval accuracy is also observed.

CommonGen. In this concept-based text generation task, HMU provides a stable improvement of +0.2–0.3 points in *accuracy*. This suggests that adaptive fusion improves the model's compositional capability by better integrating the concepts' latent meanings.

AG-News. Although the task is semantically simpler, the HMU model shows marginal but consistent improvements in precision and validation loss.

ROCStories. In the story ending selection task, both models achieve perfect accuracy (1.0). However, the model with HMU exhibits a lower **cross-entropy loss**, indicating a higher probability assignment to the correct ending, evidence of more robust narrative modeling.

4.2 Parameter Efficiency

The HMU unit introduces approximately **1 % additional** parameters compared to the base Transformer and an estimated **7 % increase in training time**. Despite this small computational cost, the benefits obtained in complex semantic and long-term memory tasks are notable.

4.3 Interpretation of Results

- **Semantic Capacity**: Linguistic-intensive tasks (e.g., CoLA, MRPC, CommonGen) particularly benefit from the VAE's latent space and adaptive gating.
- **Long-term memory**: Retention and retrieval results demonstrate that the HMU acts as an auxiliary dynamic memory, similar to the role of the hippocampus in humans.
- **Narrative stability**: The lower cross-entropy in ROCStories suggests that the model generates more confident and coherent responses in closed tasks.

Table 2: Comparison of results between standard Transformer and HMU in multiple tasks.

Task / Dataset	Metric	Transformer	HMU	Absolute
GLUE (mean)	Accuracy	0.5800	**0.5858**	+0.0058
MRPC	Accuracy	0.669	**0.689**	+0.0200
RTE	Accuracy	0.480	**0.484**	+0.0040
CoLA	Matthews Corr.	0.616	**0.691**	+0.0750
QNLI	Accuracy	0.545	**0.546**	+0.0010
SST-2	Accuracy	**0.763**	0.744	–0.0190
MNLI-m	Accuracy	**0.407**	0.361	–0.0460
Synthetic memory	Retention@200	5e-6	**1e-4**	×20
	Accuracy recovered	0.9689	**0.9733**	+0.0044
	Coherence (↓)	16.27	**15.18**	–1.09
AG-News	Accuracy	0.86225	**0.86242**	+0.00017
CommonGen	Accuracy	0.78398	**0.78613**	+0.00215
ROCStories	Accuracy	1.0000	1.0000	0
	Loss (cross-entropy)	0.02129	**0.01981**	–0.00148

5 Results Summary

- **Long-term retention:** The HMU achieves a **×20** improvement in the RETENTION@200 metric.
- **Average accuracy (GLUE):** +0.6 percentage point increase in classification with the same number of epochs.
- **Narrative:** Lower cross-entropy loss in ROCStories, indicating greater semantic security.
- **Efficiency:** Only $\sim 1\,\%$ additional parameters and $\sim 7\,\%$ extra training time.

Overall, the results support the hypothesis that a latent semantic memory, non-invasively integrated via HMU, can significantly improve contextual reasoning and text generation, with minimal overhead in terms of parameters or computation.

6 Potential Applications

The HMU module offers an efficient, modular, and easily integrable solution for extending the capabilities of Transformer models in tasks requiring semantic compression, contextual reasoning, or coherent generation. Thanks to its plug-and-play nature—that is, it can be inserted between the encoder and decoder without the need to modify or retrain its original components—it becomes an extremely versatile tool in the following contexts:

1. **Creative story and script generation.** The adaptive fusion of explicit context and latent representation allows for the generation of narrative text with greater structural coherence and thematic diversity, maintaining a fluid narrative even in long sequences.
2. **Semantic compression for chatbots with extensive context.** The HMU acts as a semantic filter that abstracts and preserves relevant information, allowing conversational assistants to maintain coherence in long dialogues without incurring excessive memory costs.
3. **Multi-turn reasoning in QA and dialogue tasks.** The module's ability to dynamically select which compressed information to reincorporate (simulating the behavior of the hippocampus) improves the consistency and depth of answers in iterative question-answering environments.
4. **Efficient and non-intrusive transfer learning.** Since the HMU does not alter the encoder or decoder layers, it can be added to pre-trained models without requiring complete retraining. This makes it an ideal solution for lightweight adaptations that require improved reasoning or retention with minimal computational cost.
5. **Modular extension for multitasking.** In contexts where a backbone encoder is shared for multiple tasks, the HMU can be adapted to each task by inserting specialized variants without duplicating the base model, facilitating a more scalable architecture.

7 Conclusions

The HMU demonstrates that a single unit functionally inspired by the human hippocampus, inserted after the Transformer encoder, can generate improvements disproportionate to its computational cost. With a parameter overhead of just $\sim 1\,\%$, substantial benefits are achieved in tasks of retention, narrative generation, and contextual comprehension.

This work opens a new line of exploration in architectures with *latent semantic memory*, where the ability to remember, abstract, and retrieve relevant information is explicitly modeled by compact and differentiable mechanisms. The adaptive fusion of current context and compressed representation allows not only to improve textual consistency but also to enhance creativity under control.

Thanks to its modular and plug-and-play nature, the HMU is easily integrated into pre-trained architectures without the need to retrain the encoder or decoder, making it especially suitable for production environments, resource-constrained systems, or *transfer learning* scenarios.

In future lines of work, the model can be extended to:

- **Multimodal models**, where the latent space acts as a common interface between language, vision, or audio.
- **Compositional and logical reasoning**, leveraging the HMU's ability to capture long-range dependencies and select relevant representations.
- **Continuous and dynamic learning**, integrating the module as a memory consolidation mechanism between episodes.

Overall, the HMU represents a step toward more interpretable, adaptive Transformers that are capable of reasoning more closely to human cognitive systems.

References

1. 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 (NeurIPS), 2017.
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019. arXiv:1810.04805.
3. Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes.* In International Conference on Learning Representations (ICLR), 2014. arXiv:1312.6114.
4. Larry R. Squire, Paul E. Garrard, and John T. Wixted. *Memory Consolidation.* Cold Spring Harbor Perspectives in Biology, 2015. 10.1101/cshperspect.a021766.
5. Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory.* Neural Computation, 9(8):1735–1780, 1997. 10.1162/neco.1997.9.8.1735.
6. Tom B. Brown, Benjamin Mann, Nick Ryder, et al. *Language Models are Few-Shot Learners.* In Advances in Neural Information Processing Systems (NeurIPS), 2020. arXiv:2005.14165.

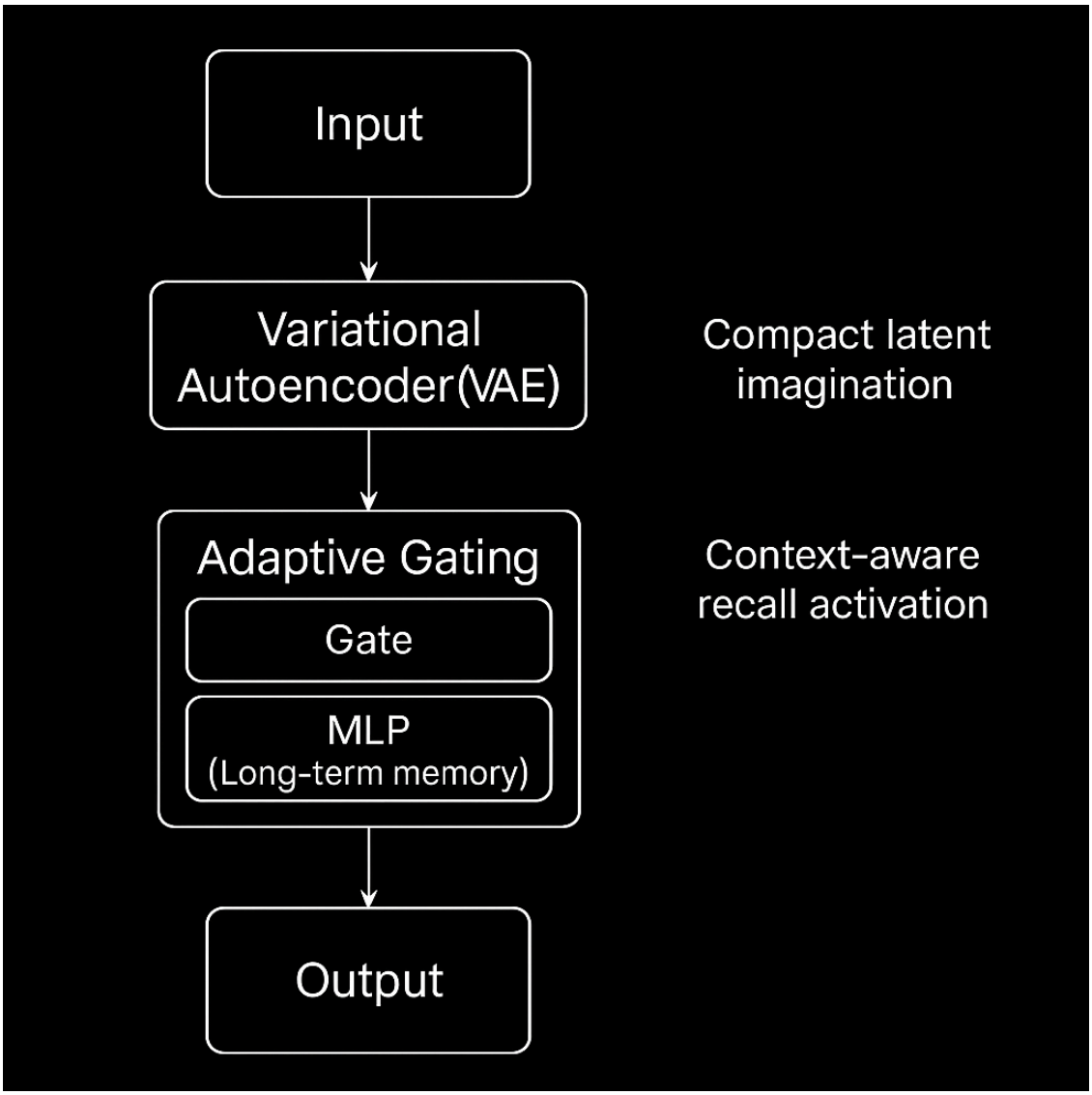

Fig. 1: Schematic diagram of the HMU.