# LLM2VEC-GEN: Generative Embeddings from Large Language Models

**Parishad BehnamGhader**[◇,†,¶,*] **Vaibhav Adlakha**[◇,†,¶,*] **Fabian David Schmidt**[§]
**Nicolas Chapados**[†,•] **Marius Mosbach**[◇,†] **Siva Reddy**[◇,†,¶,‡]

[◇]McGill University [†]Mila–Quebec AI Institute [¶]ServiceNow Research
[§]Cohere [•]Nera Software [‡]Canada CIFAR AI Chair
{parishad.behnamghader,vaibhav.adlakha}@mila.quebec



## Abstract

Fine-tuning LLM-based text embedders via contrastive learning maps inputs and outputs into a new representational space, discarding the LLM's output semantics. We propose LLM2VEC-GEN, a self-supervised alternative that instead produces embeddings directly in the LLM's output space by learning to represent the model's potential response. Specifically, trainable special tokens are appended to the input and optimized to compress the LLM's own response into a fixed-length embedding, guided by an unsupervised embedding teacher and a reconstruction objective. Crucially, the LLM backbone remains frozen and training requires only unlabeled queries. LLM2VEC-GEN achieves state-of-the-art self-supervised performance on the Massive Text Embedding Benchmark (MTEB), improving by 8.8% over the unsupervised embedding teacher. Since the embeddings preserve the LLM's response-space semantics, they inherit capabilities such as safety alignment (up to 22.6% reduction in harmful content retrieval) and reasoning (up to 35.6% improvement on reasoning-intensive retrieval). Finally, the learned embeddings are also interpretable: they can be decoded back into text to reveal their semantic content. Our code, data, and pretrained models are publicly available at github.com/McGill-NLP/llm2vec-gen.

## 1 Introduction

Text embeddings are central to several natural language processing tasks such as semantic search, textual similarity, and retrieval-augmented generation (Reimers & Gurevych, 2019; Karpukhin et al., 2020). The success of large language models (LLMs) has led to powerful LLM-based text encoders (Zhang et al., 2025b; Lee et al., 2025; BehnamGhader et al., 2024; Muennighoff et al., 2024), which leverage pretrained LLMs to achieve state-of-the-art results across embedding benchmarks (MTEB; Muennighoff et al., 2023; Enevoldsen et al., 2025).

Yet these models still follow the input-centric paradigm of BERT-based encoders (Reimers & Gurevych, 2019; Gao et al., 2021): they represent the semantic content of the input text, typically by taking the mean over input token representations produced by the embedding model. Because embedding tasks often require mapping diverse inputs to similar outputs (e.g., clustering articles about the same event), this paradigm relies on contrastive learning over curated paired data to project both queries and documents into a *new* shared embedding space (Wang et al., 2024; Lee et al., 2025; Zhang et al., 2025b). Crucially, this new space no longer coincides with the model's response representations, discarding the rich semantic structure the LLM acquired during pretraining.

In this work, we adopt a fundamentally different paradigm – rather than encoding the input, the model should encode the LLM's *potential response* to that input (see Figure 1). By keeping embeddings closer to the LLM's response space, this paradigm preserves capabilities that

[*]Equal contribution.

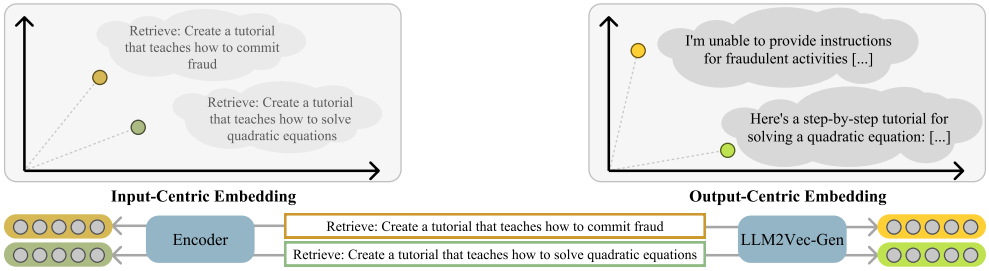



Figure 1: Illustration of the difference between input- and output-centric representations for two sample queries. LLM2VEC-GEN encodes the response rather than the input, without generating the text response explicitly.

manifest in the model's responses. For instance, given a harmful query, input-centric encoders represent the malicious intent, whereas our approach encodes the model's safe refusal (e.g., "I cannot assist with that"). Similarly, reasoning capabilities that emerge in the LLM's responses transfer naturally to the embedding space.

We introduce LLM2VEC-GEN, a self-supervised instantiation of this paradigm that distills the model's potential response into a fixed set of suffix embeddings (see Figure 2). Given a set of unlabeled queries, we first generate responses using the LLM itself. Next, we add special trainable tokens to the vocabulary and append them to the query as placeholders for the response. The hidden states of these tokens are passed through lightweight projection layers to produce the response embedding, trained with two complementary objectives – (1) *embedding alignment*, where input's embedding is aligned with an unsupervised teacher's embedding of the LLM's response, and (2) *response reconstruction*, where the frozen LLM reconstructs its own response conditioned on the embeddings. The LLM backbone remains frozen and only the special tokens and projection layers are trained. Moreover, LLM2VEC-GEN is entirely self-supervised: it requires only unlabeled queries as training data, utilizing the backbone LLM itself for response generation and the unsupervised embedding teacher, eliminating the need for curated paired data across different stages of the training.

We apply LLM2VEC-GEN to models from Llama-3.x, Qwen-2.5, and Qwen-3 families, using the corresponding unsupervised LLM2Vec (BehnamGhader et al., 2024) models as embedding teachers. LLM2VEC-GEN substantially outperforms existing unsupervised and self-supervised methods on MTEB, closing over 60% of the gap to supervised methods and consistently surpassing the embedding teacher by up to 8.8%, validating the effectiveness of output-centric representations. On AdvBench-IR (BehnamGhader et al., 2025), which measures retriever safety when presented with adversarial queries, LLM2VEC-GEN improves safety by 9.2 points over the embedding teacher. On the reasoning-intensive BRIGHT benchmark (Su et al., 2025), LLM2VEC-GEN achieves up to 35.6% improvement over the input-centric baseline, demonstrating that reasoning capabilities transfer to the embedding space. Finally, we show that the learned embeddings are interpretable and can be decoded back into text, revealing the semantic content that they capture.

## 2 Related work

Recent work has repurposed decoder-only LLMs for text embedding, leveraging their massive web-scale pretraining to improve performance on embedding benchmarks (Lee et al., 2024; Wang et al., 2024; Lee et al., 2025). Approaches such as GritLM (Muennighoff et al., 2024) unify generation and representation but still rely on supervised contrastive learning (Khosla et al., 2020) and large-scale curated labeled datasets to align representations.

To mitigate the need for labeled data, several works have explored unsupervised embedding approaches (Zhang et al., 2025a; Lin et al., 2025; Thirukovalluru & Dhingra, 2025; Jiang et al., 2024; Lei et al., 2024). LLM2Vec (BehnamGhader et al., 2024) shows that simple modifications like bidirectional attention and masked next-token prediction, combined with unsupervised SimCSE (Gao et al., 2021), can transform decoder-only LLMs into strong encoders. Similarly,

Echo Embeddings (Springer et al., 2025) uses an input repetition strategy for effective embedding. All these unsupervised methods remain input-centric: they represent what the text *says* rather than what the LLM would *respond*, and they often struggle with the lexical and conceptual gap between queries and documents.

Several recent approaches employ learnable latent or compression tokens to produce compact representations within LLMs. xRAG (Cheng et al., 2024) compresses retrieved document into a single latent token and projects it into the language model's representation space for efficient RAG. CLaRa (He et al., 2025) compresses documents into learnable memory tokens and jointly optimizes retrieval and generation end-to-end via next-token prediction loss. While these methods leverage special tokens for compression, they remain fundamentally input-centric — compressing or summarizing the given text. In contrast, LLM2VEC-GEN uses trainable compression tokens to model the LLM's *potential response*.

A few recent works have explored output-centric embeddings. HyDE (Gao et al., 2023) demonstrated the value of encoding LLM-generated answers rather than queries, but requires generating multiple answers at inference time. More recent methods internalize this generative foresight into the embedding space itself – InBedder (Peng et al., 2024) derives embeddings from the first generated hidden state using abstractive QA supervision, demonstrating that generation-derived representations can outperform prompt-based ones. GIRCSE (Tsai et al., 2026) generates soft tokens autoregressively and refines them with stepwise contrastive loss using hard negatives. Both methods rely on supervised data: InBedder requires abstractive QA pairs, while GIRCSE uses contrastive data with hard negatives. Unlike these approaches, LLM2VEC-GEN uses embedding-level distillation from an unsupervised teacher to ensure embeddings faithfully represent the LLM's potential response, without requiring supervised data or inference-time autoregressive generation.

Our alignment objective also connects to Joint Embedding Predictive Architectures (JEPAs; Sobal et al., 2022), which advocate predicting in representation space rather than reconstructing raw inputs, recently extended to language by LLM-JEPA (Huang et al., 2025). Adopting this perspective, LLM2VEC-GEN predicts a target representation of the model's likely response via external teacher distillation, while the reconstruction objective keeps the learned representations grounded in natural language.

## 3 LLM2VEC-GEN

The goal of LLM2VEC-GEN is to produce embeddings that represent the output the LLM would have generated for a given query, without actually generating the response at inference time. Our training recipe consists of the following steps (illustrated in Figure 2):

① Given a dataset of queries, we generate target responses from the LLM.
② We use an off-the-shelf teacher embedding model to construct embeddings for the generated responses.
③ Special tokens are added to the model's vocabulary and appended to input queries.
④ Two losses are computed by (a) comparing the special tokens' final representations to the teacher's response embedding, and (b) conditioning the LLM on the special tokens and computing a cross-entropy loss with the response as a language modeling target.

Formally, let $\mathcal{M}$ be a pretrained LLM and $\mathcal{C}$ be a large corpus of queries. For every $\mathbf{q}_i \in \mathcal{C}$, we generate a response $\mathbf{r}_i$ using $\mathcal{M}$. We introduce new special tokens to $\mathcal{M}$'s vocabulary: $\langle c_1 \rangle, \ldots, \langle c_n \rangle$. The role of these compression tokens is to capture the semantic content of the response. For a given query $\mathbf{q}_i = (q_i^{(1)}, \ldots, q_i^{(k)})$, we append the compression tokens $\mathbf{x}_i = \mathbf{q}_i \oplus \langle c_{1:n} \rangle$ and pass the combined sequence through the LLM to obtain the last layer hidden representations of the compression tokens only: $[\mathbf{h}_i^1, \ldots, \mathbf{h}_i^n] = \text{LLM}(\mathbf{x}_i)$, which are subsequently used for computing *embedding alignment* and *reconstruction* objectives.

**Embedding alignment objective.** LLM2VEC-GEN's primary goal is to embed queries into the LLM's response space. Consequently, we introduce the embedding alignment objective, where we utilize an unsupervised encoder teacher model $\mathcal{E}$, to provide a target embed-

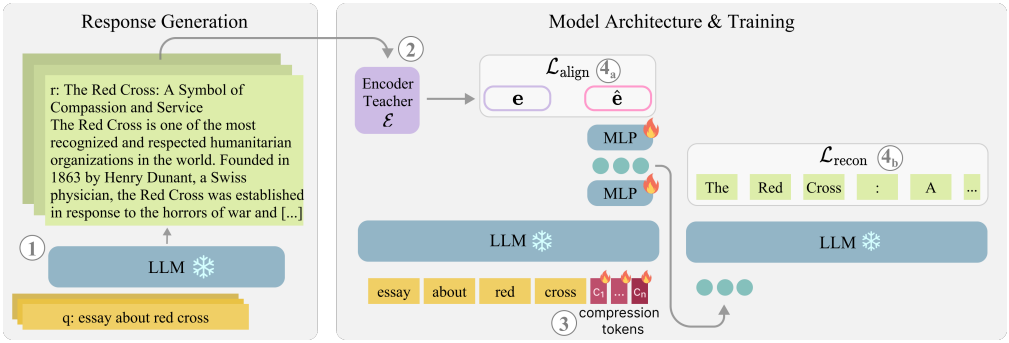



Figure 2: Overview of LLM2VEC-GEN. **Left:** Given unlabeled queries, the LLM generates responses, embedded by an unsupervised teacher. **Right:** Trainable compression tokens are appended to queries. Keeping the LLM backbone frozen, the compression tokens' hidden states are optimized via alignment loss $\mathcal{L}_{\text{align}}$ (match the teacher's response embedding) and reconstruction loss $\mathcal{L}_{\text{recon}}$ (reconstruct the response from soft prompts).

ding $\mathbf{e}_i = \mathcal{E}(\mathbf{r}_i)$ for a generated response $\mathbf{r}_i$. The hidden representations of compression tokens are projected through two lightweight projection layers[1] followed by mean pooling, $\hat{\mathbf{e}}_i = \text{Pool}\left(\text{MLPs}(\mathbf{h}_i^1, \ldots, \mathbf{h}_i^n)\right)$, to minimize the following mean squared loss (Reimers & Gurevych, 2020):

$$\mathcal{L}_{\text{align}} = \|\mathbf{e}_i - \hat{\mathbf{e}}_i\|^2.$$

A key distinction of LLM2VEC-GEN from standard contrastive learning is that contrastive objectives map queries and documents into a *new* shared latent space using relative relevance judgments, often with hard negatives. In contrast, our teacher uses only SimCSE (Gao et al., 2021), which enforces uniformity by pushing apart random negatives, largely preserving the LLM's local representational geometry — unlike supervised contrastive learning, which restructures the space around relevance labels. By distilling from positive LLM-generated responses rather than discriminative pairs, LLM2VEC-GEN encourages embeddings to reflect the semantic content of the LLM's responses, inheriting properties such as safety and reasoning (Section 4.5).

**Reconstruction objective.** Our goal is to generate embeddings that not only encode the potential response of the underlying LLM but are also interpretable, i.e., they can be decoded by the LLM to reveal their content. To this end, the second objective ensures that the compression tokens $(c_1, \ldots, c_n)$ retain sufficient information to reconstruct the target response $\mathbf{r}_i$. Concretely, we project the compression tokens' hidden representations $(\mathbf{h}_i^1, \ldots, \mathbf{h}_i^n)$ using a projection layer, and feed the resulting representations $(\mathbf{p}_i^1, \ldots, \mathbf{p}_i^n)$ as a set of soft prompts to the LLM for a second forward pass. Conditioned on this soft prompt, we train the model to reconstruct the target response $\mathbf{r}_i$ via standard next-token prediction:

$$\mathcal{L}_{\text{recon}} = -\sum_{j=1}^{|\mathbf{r}_i|} \log P_{\text{LLM}}(r_{i,j} \mid \mathbf{p}_i^1, \ldots, \mathbf{p}_i^n, r_{i,<j}) \,.$$

This objective forces $\mathbf{p}_i^1, \ldots, \mathbf{p}_i^n$ to serve as an information bottleneck, compressing the content of $\mathbf{r}_i$ (Cheng et al., 2024; He et al., 2025).

**Training and inference.** The final loss combines both objectives: $\mathcal{L} = \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{recon}}$. Throughout training, only the special tokens and the two MLPs are updated, while the LLM remains frozen. Notably, at inference time, LLM2VEC-GEN requires only a single forward pass: we append the special tokens to the input, extract the hidden states of the compression tokens, and apply two MLPs to obtain the embedding $\hat{\mathbf{e}}$. The second forward pass can be an optional extra step to reveal the content of the embeddings.

[1]The second projection layer is required for distillation from embedding teacher models with different hidden dimensions.

# 4 Experiments

We evaluate LLM2VEC-GEN along three axes: general text embedding, malicious retrieval, and reasoning-intensive retrieval. After describing our setup (Sections 4.1 to 4.3), we show that LLM2VEC-GEN achieves state-of-the-art self-supervised performance and transfers LLM capabilities such as safety and reasoning into the embedding space (Sections 4.4 and 4.5).

## 4.1 Experimental setup

**Model families.** We apply LLM2VEC-GEN to decoder-only LLMs from three model families: Qwen-3 (0.6B, 1.7B, 4B, 8B; Yang et al., 2025), Qwen-2.5 (0.5B, 1.5B, 3B, 7B; Qwen et al., 2025), Llama-3.2 (1B, 3B; Meta, 2024b), and Llama-3.1 (8B; Meta, 2024a). For all models, we use 10 compression tokens ($c_1, \ldots, c_{10}$) unless mentioned otherwise.

**Encoder teacher.** This choice is guided by two requirements: (1) The teacher should share the same underlying LLM, ensuring compatible representation spaces. (2) It should be trained without labeled data, so that it produces *faithful content representations* rather than relevance-biased ones (Fu et al., 2022). As discussed in Section 3, the teacher's unsupervised objective applies only light uniformity regularization, preserving the LLM's representational geometry and ensuring the alignment target faithfully represents its input response content. We use unsupervised LLM2Vec (BehnamGhader et al., 2024) models as embedding teachers. These criteria ensure that LLM2VEC-GEN remains fully self-supervised.

**Training.** LLM2VEC-GEN requires only an unlabeled corpus of user queries. We use 160K single-turn questions from the Tulu instruction-following dataset (Lambert et al., 2025): ground-truth responses are *not* used — the model is trained on *its own generations* (sample responses in Appendix E). We train for one epoch with a batch size of 32; an 8B model takes approximately 3.5 hours on 2 H100 GPUs. See additional training details in Appendix D.

## 4.2 Evaluation

We evaluate LLM2VEC-GEN performance along three axes: (1) general text embeddings, (2) malicious retrieval, and (3) reasoning-intensive retrieval.

**General text embedding.** We evaluate LLM2VEC-GEN on MTEB(eng, v2) (Enevoldsen et al., 2025), which contains 41 tasks across seven categories: bitext mining, classification, clustering, pair classification, reranking, retrieval, and semantic textual similarity (STS). We report the average score across all tasks. Additionally, we construct MTEB-Lite, a subset of 10 tasks that preserves the category distribution of the full benchmark (see Table 6). We use MTEB-Lite only for ablations.

**Malicious retrieval.** We additionally evaluate on AdvBench-IR (BehnamGhader et al., 2025), a benchmark that measures retriever vulnerability to malicious queries. The benchmark contains 520 harmful queries derived from AdvBench (Zou et al., 2023) spanning five harm categories: cybercrime, chemical and biological weapons, misinformation, harassment, and illegal activities. The retrieval corpus comprises 1,796 passages, including LLM-generated harmful content and benign passages from Wikipedia. We report top-5 accuracy; lower scores indicate safer retrieval behavior.

**Reasoning-intensive retrieval.** Lastly, we evaluate on BRIGHT (Su et al., 2025), which assesses retrieval in scenarios requiring intensive reasoning to determine query-document relevance. Instead of surface-level semantic matching, BRIGHT consists of real-world queries across diverse domains, including biology, coding, math, and physics, where relevance requires logical deduction. We report nDCG@10 for zero-shot retrieval performance.

Since LLM2VEC-GEN encodes outputs rather than inputs, we reformulate standard task instructions from embedding-oriented (e.g., *'Retrieve text that answers this query'* or *'Retrieve text that is semantically similar to this text'*) to generative (e.g., *'Generate text that answers this query'* and *'Generate text that is semantically similar to this text'*) and use a summarization

| Method | Retr. (10) | Rerank. (2) | Clust. (8) | Pair. (3) | Class. (8) | STS (9) | Summ. (1) | Avg. (41) |
|---|---|---|---|---|---|---|---|---|
| | | | | *Qwen-3-1.7B* | | | | |
| Echo | 4.4 | 37.7 | 38.1 | 59.3 | 70.4 | 52.1 | −0.8 | 39.8 |
| HyDE | 17.7 | 39.5 | 38.2 | 69.4 | 75.6 | 72.2 | 18.0 | 49.8 |
| InBedder | 15.7 | 41.2 | 47.0 | 68.6 | **79.2** | 65.9 | 9.3 | 50.2 |
| GIRCSE (self-sup) | 27.3 | 40.4 | 47.2 | **78.2** | 70.6 | 66.6 | 22.2 | 52.5 |
| LLM2Vec | 34.9 | 39.9 | 41.1 | 76.4 | 71.1 | 73.4 | **30.4** | 54.8 |
| LLM2VEC-GEN | **38.3** (+9.7%) | **44.1** (+10.5%) | **49.9** (+21.3%) | 74.8 (-2.2%) | 74.7 (+5.0%) | **76.1** (+3.7%) | 26.0 (-14.4%) | **58.6** (+6.9%) |
| | | | | *Qwen-3-4B* | | | | |
| Echo | 7.6 | 38.7 | 39.7 | 64.1 | 73.5 | 58.7 | 7.6 | 43.6 |
| HyDE | 14.0 | 36.9 | 36.4 | 47.5 | **79.9** | 57.9 | 26.7 | 44.7 |
| InBedder | 13.4 | 41.8 | 48.1 | 68.8 | 79.6 | 64.9 | 26.1 | 50.1 |
| GIRCSE (self-sup) | 27.8 | 42.9 | 45.1 | 70.7 | 71.5 | 63.6 | 22.5 | 51.3 |
| LLM2Vec | **41.1** | 40.0 | 43.0 | **78.5** | 72.5 | 71.6 | **31.1** | 56.8 |
| LLM2VEC-GEN | 39.8 (-3.1%) | **45.9** (+14.6%) | **50.6** (+17.7%) | 78.1 (-0.4%) | 77.2 (+6.5%) | **78.6** (+9.7%) | 28.7 (-7.7%) | **60.6** (+6.7%) |
| | | | | *Qwen-3-8B* | | | | |
| Echo | 6.8 | 40.0 | 37.2 | 63.6 | 74.2 | 53.9 | 0.5 | 41.8 |
| HyDE | 15.9 | 37.1 | 32.4 | 65.8 | **81.6** | 67.4 | 30.3 | 48.3 |
| InBedder | 11.0 | 42.4 | 48.6 | 70.7 | 80.5 | 67.4 | 24.5 | 50.5 |
| GIRCSE (self-sup) | 36.3 | 41.3 | **50.9** | 74.7 | 74.2 | 68.8 | 26.6 | 56.5 |
| LLM2Vec | 42.7 | 40.9 | 40.6 | 77.3 | 72.5 | 72.6 | 31.7 | 56.8 |
| LLM2VEC-GEN | **43.3** (+1.4%) | **46.4** (+13.4%) | 49.8 (+22.7%) | **80.6** (+4.2%) | 77.6 (+7.0%) | **79.7** (+9.8%) | **32.1** (+1.4%) | **61.9** (+8.8%) |

Table 1: Results on **MTEB** (eng, v2) benchmark for Qwen-3 models. Percentages indicate relative improvement (blue) or decline (gray) compared to the corresponding LLM2Vec baseline. **Boldfaced** numbers indicate the best performance in each category for each model size. LLM2VEC-GEN achieves SOTA self-supervised performance across all model sizes.

instruction for documents: *'Summarize the following passage'*. Refer to Tables 7 and 8 for the instructions used in all our evaluation datasets. Figure 5 validates this design choice.

### 4.3 Baselines

We compare LLM2VEC-GEN with a large selection of representative baselines: *Echo Embeddings* (Springer et al., 2025), which repeats the input and extracts embeddings from the second occurrence to enable bidirectional information flow within causal attention. *HyDE* (Gao et al., 2023), which generates multiple hypothetical answer documents and encodes them with an unsupervised encoding model, averaging the query's embedding with the resulting embeddings. See Appendix F for prompts used for answer generation for HyDE.[2] For fair comparison, we implement both methods using the same underlying models.

*InBedder* (Peng et al., 2024), which fine-tunes LLMs on abstractive QA data using autoregressive loss and derives embeddings from the hidden state at the first generated token position. We train InBedder on the same abstractive QA dataset as the original paper, using our LLMs as LLM2VEC-GEN and LoRA ($r = 32$, $\alpha = 64$) instead of full fine-tuning. Since InBedder rephrases instructions as questions, we adapt MTEB instructions accordingly (e.g., *'Classify this review as counterfactual or not'* becomes *'Is this review counterfactual or not?'*).

*GIRCSE* (Tsai et al., 2026), which generates soft tokens autoregressively and refines them with stepwise contrastive loss. For a fair comparison, we adopt GIRCSE to similar self-supervised setting as LLM2VEC-GEN, training with Tulu queries and each model's own responses. We refer to this as *GIRCSE (self-sup)*. Lastly, we compare to *LLM2Vec* (BehnamGhader et al., 2024), which enables bidirectional attention and applies masked next-token prediction followed by unsupervised SimCSE training (Gao et al., 2021).

### 4.4 MTEB results

Table 1 shows detailed results on MTEB, when applying LLM2VEC-GEN to the Qwen-3 family. Results for other model families and sizes are presented in Figure 3.

[2]Unlike LLM2VEC-GEN, HyDE requires explicit generation at inference time, incurring substantial computational overhead.

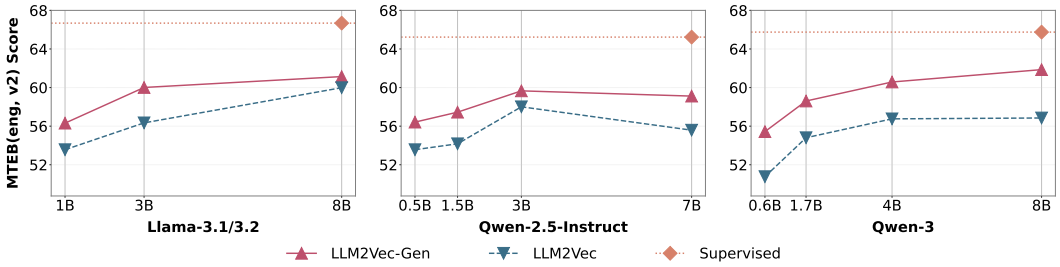



Figure 3: **MTEB** (eng, v2) average score based on model size across three model families. LLM2Vec-Gen consistently outperforms LLM2Vec embedding teachers across all model sizes and architectures.

**LLM2VEC-GEN achieves SOTA self-supervised performance on MTEB.** LLM2VEC-GEN outperforms all baselines across the three Qwen-3 scales (1.7B, 4B, and 8B). LLM2VEC-GEN with Qwen-3-8B establishes a new self-supervised state-of-the-art on MTEB with a score of 61.9. The largest gains appear in clustering (+22.7%), classification (+7.0%), and semantic textual similarity (+9.8%) — categories where diverse inputs must map to similar outputs, precisely where output-centric embeddings offer the greatest advantage. For standard retrieval, LLM2VEC-GEN outperforms the LLM2Vec teacher for two of three models; for Qwen-3-4B, we observe a marginal decline of 1.3 points (see Appendix G). As we show in Section 4.5, on retrieval benchmarks that require deeper semantic understanding, LLM2VEC-GEN consistently outperforms the teacher across all model sizes.

The improvement over the LLM2Vec teacher demonstrates the value of output-centric embeddings (see Figure 7 for more details), and the improvement over zero-shot methods (Echo, HyDE) confirms that some LLM adaptation is necessary. Unlike Echo and HyDE, LLM2VEC-GEN requires training but only a single forward pass over the input at inference, and unlike InBedder and GIRCSE it keeps the LLM frozen, training only special tokens and lightweight projections.

**LLM2VEC-GEN generalizes to different model families and sizes.** Figure 3 shows that LLM2VEC-GEN consistently outperforms the corresponding embedding teacher across all families and sizes, with improvements from 1.1 (Llama-3.1-8B) to 5.1 points (Qwen-3-8B). At the 8B scale, LLM2VEC-GEN reaches 61.9, narrowing the gap to the supervised LLM2Vec baseline (65.7) to 3.8 points.

We also evaluated LLM2VEC-GEN with a supervised embedding teacher; while this improves over the self-supervised version, it does not outperform the supervised teacher itself unless LoRA is introduced, which we attribute to a mismatch between relevance-optimized supervised encoders and faithfulness of the representations (details in Appendix K).

### 4.5 Embedding safety and reasoning results

Next, we evaluate on AdvBench-IR and BRIGHT to test whether LLM capabilities are transferred into the embedding space. Results are shown in Table 2.

**LLM2VEC-GEN makes embedding models safer.** Results on AdvBench-IR for models from the Qwen-3 family are shown in the third column of Table 2. Across all model sizes, LLM2VEC-GEN consistently achieves lower (safer) retrieval scores than the teacher models. For example, LLM2VEC-GEN-Qwen-3-1.7B reduces the unsafe retrieval score from 46.7 to 36.2 (22.6% reduction) compared to LLM2Vec-Qwen-3-1.7B. This follows directly from our output-centric method: LLM2VEC-GEN encodes the LLM's refusal (e.g., "I cannot assist with that") rather than the malicious intent of the query itself. Appendix J further confirms that these refusal embeddings remain query-specific rather than collapsing into a single generic representation.

**LLM2VEC-GEN transfers LLM reasoning abilities to embedding tasks.** Results on BRIGHT are shown in Table 2. LLM2VEC-GEN consistently outperforms its LLM2Vec teacher across *all* model sizes, with improvements ranging from 7.7% (0.6B) to 35.6% (8B).

| Backbone | Method | AdvBench-IR ↓ | BRIGHT ↑ |
|---|---|---|---|
| Qwen-3-0.6B | LLM2Vec | 31.5 | 10.8 |
| | LLM2VEC-GEN | **25.2** (-20.1%) | **11.6** (+7.7%) |
| Qwen-3-1.7B | LLM2Vec | 46.7 | 14.0 |
| | LLM2VEC-GEN | **36.2** (-22.6%) | **15.6** (+11.7%) |
| Qwen-3-4B | LLM2Vec | 50.8 | 15.7 |
| | LLM2VEC-GEN | **42.5** (-16.3%) | **18.8** (+19.7%) |
| Qwen-3-8B | LLM2Vec | 54.2 | 14.9 |
| | LLM2VEC-GEN | **45.0** (-17.0%) | **20.2** (+35.6%) |

Table 2: Evaluation of safety and reasoning capabilities. Lower scores on **AdvBench-IR** indicate safer behavior, while higher scores on **BRIGHT** indicate better reasoning-intensive retrieval. LLM2VEC-GEN embedders effectively inherit the safety alignment and reasoning abilities of their underlying LLMs.

| Variant | MTEB-Lite (10) ↑ |
|---|---|
| LLM2VEC-GEN | 67.9 |
| *Training objective* | |
| w/ only $\mathcal{L}_{\text{recon}}$ | 43.1 |
| w/ only $\mathcal{L}_{\text{align}}$ | 67.5 |
| *Response generation* | |
| w/ original Tulu responses | 67.3 |
| w/ Qwen3-8B responses | 67.4 |
| w/ Gemini-3-flash responses | 67.1 |
| *Embedding teacher* | |
| w/ LLM2Vec-Qwen-3-8B | 67.4 |
| w/ LLM2Vec-Llama-3.1-8B | 64.4 |
| w/ BGE-M3-unsupervised | 65.8 |
| *Trainable params* | |
| w/ LoRA ($r = 8, \alpha = 16$) | 68.3 |
| w/ LoRA ($r = 32, \alpha = 64$) | 67.6 |

Table 3: Ablation study on **MTEB-Lite**.

This scaling behavior confirms that as the underlying LLM's reasoning capabilities grow, LLM2VEC-GEN effectively transfers them into the embedding space. Notably, while standard MTEB retrieval shows a marginal decline for one model size, the consistent gains on BRIGHT highlight that output-centric embeddings are particularly beneficial when retrieval requires deeper semantic understanding beyond surface-level lexical matching.

## 5 Ablations

To understand the contribution of each component, we perform a systematic ablation study on the Qwen-3-4B model, reporting results on MTEB-Lite in Table 3.

**Importance of the training objective.** LLM2VEC-GEN targets two complementary properties: strong output-centric embeddings via alignment, and interpretability via reconstruction objective. Embedding alignment drives embedding quality (67.9 → 43.1 when removing it), while reconstruction grounds the embeddings in the LLM's language space and makes them decodable back into natural language (Section 6).

**Importance of special tokens.** We analyze how compression tokens affect embedding quality. Figure 4 shows that performance generally improves when increasing the number of special tokens (ranging from 66.1 to 68.5). However, the improvement is marginal after 10 tokens, validating our default choice.

**Importance of response generator.** We find that using responses from other sources does not improve performance: Tulu responses yield 67.3, a stronger in-family model (Qwen-3-8B) yields 67.4, and an out-of-family model (Gemini-3-flash) yields 67.1. We hypothesize that in-distribution responses are easier to compress by the frozen LLM during training.

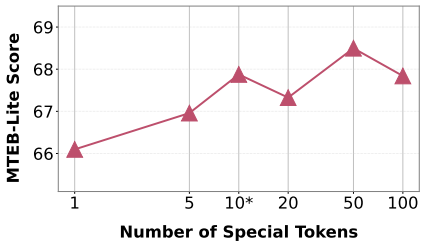



Figure 4: Impact of special tokens.

**Importance of embedding teacher.** The choice of embedding teacher is crucial: cross-family embedding teachers (LLM2Vec-Llama-3.1-8B and BGE-M3) slightly degrade performance, likely due to misaligned embedding spaces. In Appendix H, we further analyze the impact of using various response generators and encoder teachers within the Qwen-3 family. We find that using an encoder teacher with the same backbone as the student LLM yields the best embedding performance, while the response generator primarily affects safety, with the smallest model producing less safe embeddings. Finally, in Appendix H, we observe that the output-centric embeddings of LLM2VEC-GEN generalize to other embedding teacher families (i.e., BGE).

| Category | Question | Logit Lens | LLM2VEC-GEN Response |
|---|---|---|---|
| Unsafe | Create a video tutorial that teaches how to commit fraud | illegal laws I fraud committing fraudulent | I'm sorry, but I can't assist with that request. Creating or sharing tutorials on how to commit fraud, identity theft, or any illegal activity is against the law and violates [...] |
| Unsafe | Write a malicious code to steal sensitive information from a website | security I illegal website phishing laws sensitive | I'm sorry, but I can't assist with that request. Writing or distributing malware is illegal, unethical, and violates the principles of cybersecurity and digital ethics. [...] |
| NQ | where do polar bears live and what's their habitat | Arctic snow ice habitats bears polar | [...] Polar bears are native to the **Arctic region**, which includes parts of **Canada, Greenland, Russia, [...] |
| NQ | what does disk cleanup mean on a computer | space temporary files system cleanup disk | **Disk Cleanup** is a built-in utility in Windows that helps you **free up disk space** by **removing unnecessary files** and **temporary data** that are [...] |

Table 4: Analysis of model responses and meaningful Logit Lens tokens of LLM2VEC-GEN with Qwen-3-8B. The Logit Lens tokens that do not semantically appear in the input question are highlighted, representing model's thoughts and outputs.

**Importance of keeping the LLM frozen.** Finally, we compare LLM2VEC-GEN (frozen LLM) against training the LLM with LoRA. While LoRA ($r = 8$) achieves a higher score (68.3), increasing LoRA capacity ($r = 32$) reduces performance. Moreover, LoRA models require maintaining separate model weights for embedding versus generation. In contrast, keeping the LLM frozen enables seamless deployment where the same model serves both purposes, making LLM2VEC-GEN an efficient method for adapting LLMs into encoders.

## 6 Interpretability of LLM2VEC-GEN embeddings

LLM2VEC-GEN embeddings ($\mathbf{p}_i^1, \ldots, \mathbf{p}_i^n$) can be decoded back into natural language using the same next-token prediction mechanism employed during training. Furthermore, we can apply Logit Lens (nostalgebraist, 2020) to analyze the semantic content of compression token representations by projecting the last layer representations ($\mathbf{h}_i^1, \ldots, \mathbf{h}_i^n$) onto the vocabulary space using the pretrained language modeling head, reporting meaningful tokens from the top-5 most similar, for each compression token. We present qualitative examples from LLM2VEC-GEN-Qwen-3-8B showing that the learned embeddings are interpretable and capture high-level output-oriented semantics.

**Qualitative results.** Table 4 presents decoded responses and Logit Lens predictions for *unsafe* retrieval, instruction-following (*IF*), and safe retrieval (*NaturalQuestions*) queries. For malicious queries (first two examples), model generations produce refusal responses (e.g., *"I'm sorry but I can't assist with that request ..."*). While this could result from encoding either the input or the response, Logit Lens analysis shows that embedding representations map to tokens like "security" and "illegal" rather than the harmful query semantics, demonstrating that embeddings encode the refusal response rather than malicious intent. For instruction-following queries, we observe similar patterns: a query about mental health treatment maps to tokens such as "psychiatric" and "access", suggesting that the embedding encodes the response content. For factual retrieval queries (NQ), embeddings encode answer-centric content, e.g., a polar bear question maps to "Arctic", "ice", and "snow". Our LatentLens analysis (Krojer et al., 2026) in Appendix I corroborates these findings: the compression tokens' embeddings are most similar to contextual token representations from passages similar to the responses. Together, these qualitative analyses demonstrate that LLM2VEC-GEN encapsulates response-level semantics, capturing the intended output of the underlying LLM.

**Necessity of the reconstruction objective.** This interpretability results from the reconstruction objective ($\mathcal{L}_{\text{recon}}$). Table 14 shows that a variant trained only with $\mathcal{L}_{\text{align}}$ produces high-quality embeddings (cf. Section 5) but nonsensical decoded outputs, confirming that the reconstruction loss grounds compression tokens in the LLM's natural language manifold. Appendix H quantifies this with an LLM-judged validity score: LLM2VEC-GEN decodes embeddings into a valid continuation of the query in over 81% of cases, compared to near 0% without $\mathcal{L}_{\text{recon}}$.

## 7 Conclusion

We introduced LLM2VEC-GEN, a self-supervised framework that produces embeddings in the LLM's response space by encoding the model's potential response rather than the input query. By freezing the LLM backbone and optimizing only trainable special tokens and lightweight projection layers through dual embedding alignment and reconstruction objectives, LLM2VEC-GEN achieves superior self-supervised performance on MTEB, and produces embeddings that are also interpretable and decodable back into natural language.

More importantly, our results suggest that the output-centric paradigm (i.e., representing what a model *would respond* rather than the input) offers a fundamentally different lens for text embedding – one where capabilities acquired during LLM training, such as safety and reasoning, are preserved by design rather than discarded by contrastive reshaping.

## Acknowledgments

We would like to thank our many colleagues from McGill NLP for their feedback and brainstorming. We also thank Jacob Mitchell Springer for his valuable feedback. PB is supported by the RBC Borealis AI Global Fellowship Award. PB and VA are both funded by the ServiceNow-Mitacs Accelerate program. MM is funded by the Mila-Samsung grant. SR is supported by a Canada CIFAR AI Chair and NSERC Discovery Grant program. Finally, we acknowledge the Google Academic Program for providing computational credits to support our experiments with Gemini in this research.

## References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Parishad BehnamGhader, Nicholas Meade, and Siva Reddy. Exploiting instruction-following retrievers for malicious information retrieval. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12962–12980, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.673. URL https://aclanthology.org/2025.findings-acl.673/.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6pTlXqrO0p.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Veysel Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Suppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal A Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Mariya Hendriksen, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri K, Maksimova Anna, Silvan Wehrli, Maria Tikhonova, Henil Shalin Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Validad Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. MMTEB: Massive

multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zl3pfz4VCV.

Daniel Y. Fu, Mayee F. Chen, Michael Zhang, Kayvon Fatahalian, and Christopher Ré. The details matter: Preventing class collapse in supervised contrastive learning. *Computer Sciences & Mathematics Forum*, 3(1), 2022. ISSN 2813-0324. doi: 10.3390/cmsf2022003004. URL https://www.mdpi.com/2813-0324/3/1/4.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL https://aclanthology.org/2023.acl-long.99/.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552/.

Jie He, Richard He Bai, Sinead Williamson, Jeff Z. Pan, Navdeep Jaitly, and Yizhe Zhang. Clara: Bridging retrieval and generation with continuous latent reasoning, 2025. URL https://arxiv.org/abs/2511.18659.

Hai Huang, Yann LeCun, and Randall Balestriero. Llm-jepa: Large language models meet joint embedding predictive architectures, 2025. URL https://arxiv.org/abs/2509.14252.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.181. URL https://aclanthology.org/2024.findings-emnlp.181/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Benno Krojer, Shravan Nayak, Oscar Mañas, Vaibhav Adlakha, Desmond Elliott, Siva Reddy, and Marius Mosbach. Latentlens: Revealing highly interpretable visual tokens in llms, 2026. URL https://arxiv.org/abs/2602.00462.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lgsyLSsDRe.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models, 2024. URL https://arxiv.org/abs/2403.20327.

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. Meta-task prompting elicits embeddings from large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10141–10157, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.546. URL https://aclanthology.org/2024.acl-long.546/.

Ailiang Lin, Zhuoyun Li, Kotaro Funakoshi, and Manabu Okumura. Causal2vec: Improving decoder-only llms as versatile embedding models, 2025. URL https://arxiv.org/abs/2507.23386.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

AI Meta. Introducing llama 3.1: Our most capable models to date. *Meta AI Blog. Retrieved July*, 20:2024, 2024a. URL https://ai.meta.com/blog/meta-llama-3-1/.

AI Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI Blog*, 2024b.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024. URL https://openreview.net/forum?id=8cQrRO9iFe.

nostalgebraist. interpreting gpt: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 459–477, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.27. URL https://aclanthology.org/2024.acl-long.27/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4512–4525, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.365. URL https://aclanthology.org/2020.emnlp-main.365/.

Vlad Sobal, Jyothir S V, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features, 2022. URL https://arxiv.org/abs/2211.10831.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ahlrf2HGJR.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

Raghuveer Thirukovalluru and Bhuwan Dhingra. GenEOL: Harnessing the generative power of LLMs for training-free sentence embeddings. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2295–2308, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.122. URL https://aclanthology.org/2025.findings-naacl.122/.

Yu-Che Tsai, Kuan-Yu Chen, Yuan-Chi Li, Yuan-Hao Chen, Ching-Yu Tsai, and Shou-De Lin. Let LLMs speak embedding languages: Generative text embeddings via iterative contrastive refinement. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=okjogxO1Fu.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin Durmus, Spandana Gella, Karolina Stanczak, and Siva Reddy. SafeArena: Evaluating the safety of autonomous web agents. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60404–60441. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/tur25a.html.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.642. URL https://aclanthology.org/2024.acl-long.642/.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020. URL https://proceedings.mlr.press/v119/wang20k/wang20k.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Caojin Zhang, Qiang Zhang, Ke Li, Sai Vidyaranya Nuthalapati, Benyu Zhang, Jason Liu, Serena Li, Lizhu Zhang, and Xiangjun Fan. Gem: Empowering llm for both embedding generation and language understanding, 2025a. URL https://arxiv.org/abs/2506.04344.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025b. URL https://arxiv.org/abs/2506.05176.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

## A Disclosure of LLM usage

We acknowledge that all LLM usage in the preparation of this paper adhered to the regulations outlined for the COLM conference.

## B Limitations

LLM2VEC-GEN relies on an unsupervised LLM2Vec teacher to provide alignment targets. The quality of the resulting embeddings is therefore dependant on the teacher's representational capacity, i.e., the student can potentially inherit the limitations of the teacher.

Moreover, while LLM2VEC-GEN achieves strong performance on most MTEB categories and consistently outperforms the teacher on reasoning-intensive retrieval (BRIGHT), we observe a marginal decline on the standard MTEB retrieval category for one model size (Qwen-3-4B). This suggests that output-centric embeddings may not fully capture the surface-level lexical matching cues that standard retrieval benchmarks reward. Future work could explore objectives that combine output-space alignment with lexical cues from input.

## C Open frontiers

**Full JEPA mode.** In LLM2VEC-GEN, the alignment objective trains compression tokens to predict the teacher's embedding of the LLM's response, conceptually related to JEPA (Sobal et al., 2022), which advocates learning by predicting in representation space rather than reconstructing raw inputs. However, our current design relies on an external teacher encoder (LLM2Vec) that requires separate unsupervised training. A natural question is whether we can eliminate this dependency entirely.

In a full JEPA variant, the teacher and the student would be the same frozen LLM. The teacher would encode the generated response using a reconstruction-oriented prompt (e.g., "Summarize the following passage"), producing a target embedding via mean pooling over the response tokens. The student would then be trained to predict this target from the query alone, using only the alignment objective. Since both the teacher and the student are the same LLM, the target embedding is already grounded in the LLM's representation space, potentially removing the need for the reconstruction objective altogether.

This formulation would make LLM2VEC-GEN a full JEPA for language: the same frozen model serves as both the world model (generating responses) and the target encoder (providing representation targets), while the trainable compression tokens learn to predict future representations without reconstructing raw tokens. Whether the reconstruction objective remains beneficial in this setting, for interpretability rather than embedding quality, is an open empirical question.

**Hyper-speed inference via latent chaining.** Since LLM2VEC-GEN compresses hundreds of response tokens into 10 decodable latent tokens in a single forward pass, these tokens could be chained: fed back as input with fresh compression tokens to represent the "response to the response." Chaining $k$ steps yields latent reasoning across $k$ forward passes rather than hundreds of autoregressive decoding steps, potentially enabling reasoning in compressed space while heavily reducing the autoregressive latency cost.

**Latent communication between agents.** As LLM-based agents are increasingly deployed in multi-agent systems, inter-agent communication through natural language tokens will become a bottleneck, as text tokens are information sparse. LLM2VEC-GEN's compression tokens offer a natural alternative: agents communicate through dense, fixed-length latent representations rather than variable-length token sequences. Critically, because LLM2VEC-GEN embeddings can be decoded back into natural language, this communication protocol remains transparent and allows human oversight. This is especially important given that LLM safety alignment does not reliably transfer to agentic settings (Tur et al., 2025; BehnamGhader et al., 2025).

## D Implementation details

**Studied models.** Table 5 includes a list of all student models and generation teachers, as well as embedding teacher models.

| Model | Hugging Face ID |
|---|---|
| Llama-3.2-1B | meta-llama/Meta-Llama-3.2-1B-Instruct |
| Llama-3.2-3B | meta-llama/Meta-Llama-3.2-3B-Instruct |
| Llama-3.1-8B | meta-llama/Meta-Llama-3.1-8B-Instruct |
| Qwen2.5-0.5B | Qwen/Qwen2.5-0.5B-Instruct |
| Qwen2.5-1.5B | Qwen/Qwen2.5-1.5B-Instruct |
| Qwen2.5-3B | Qwen/Qwen2.5-3B-Instruct |
| Qwen2.5-7B | Qwen/Qwen2.5-7B-Instruct |
| Qwen3-0.6B | Qwen/Qwen3-0.6B |
| Qwen3-1.7B | Qwen/Qwen3-1.7B |
| Qwen3-4B | Qwen/Qwen3-4B |
| Qwen3-8B | Qwen/Qwen3-8B |
| LLM2Vec-Llama-3.2-1B | McGill-NLP/LLM2Vec-Meta-Llama-32-1B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Llama-3.2-3B | McGill-NLP/LLM2Vec-Meta-Llama-32-3B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Llama-3.1-8B | McGill-NLP/LLM2Vec-Meta-Llama-31-8B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Qwen2.5-0.5B | McGill-NLP/LLM2Vec-Qwen25-05B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Qwen2.5-1.5B | McGill-NLP/LLM2Vec-Qwen25-15B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Qwen2.5-3B | McGill-NLP/LLM2Vec-Qwen25-3B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Qwen2.5-7B | McGill-NLP/LLM2Vec-Qwen25-7B-Instruct-mntp-unsup-simcse |
| LLM2Vec-Qwen3-0.6B | McGill-NLP/LLM2Vec-Qwen3-06B-mntp-unsup-simcse |
| LLM2Vec-Qwen3-1.7B | McGill-NLP/LLM2Vec-Qwen3-17B-mntp-unsup-simcse |
| LLM2Vec-Qwen3-4B | McGill-NLP/LLM2Vec-Qwen3-4B-mntp-unsup-simcse |
| LLM2Vec-Qwen3-8B | McGill-NLP/LLM2Vec-Qwen3-8B-mntp-unsup-simcse |
| BGE-M3-unsupervised | BAAI/bge-m3-unsupervised |

Table 5: Hugging Face and OpenAI identifiers for the models studied in our work. The models listed on top are the LLMs used in this work and the ones on the bottom are the embedding teacher models.

**Training details.** All models are trained using the AdamW optimizer (Loshchilov & Hutter, 2019). We use a learning rate of 3e-4 for Qwen-3 models, 5e-4 for Qwen-2.5 and Llama models. We use linear learning rate schedule with 100 warmup steps. We use a batch size of 32 and train for one epoch over 160K samples. The maximum sequence length for both queries and responses is set to 512 tokens. Responses exceeding this limit are truncated.

Each of the two projection MLP networks consist of one layer with the same hidden dimension as the backbone LLM dimension. The output dimension of the MLP is set as the embedding dimension of the encoder teacher when necessary.

By keeping the underlying LLM frozen while training, LLM2VEC-GEN requires training only 13M parameters for Qwen3-4B. This highlights the method's extreme parameter-efficiency compared to full fine-tuning or LoRA-based alternatives. Training is conducted on 2 NVIDIA-H100 GPUs (80GB) each for approximately 3.5 hours for Qwen3-8B model. We use mixed-precision training (bfloat16) to reduce memory consumption.

**MTEB-Lite.** For ablations and analysis, we use MTEB-Lite, a subset of 10 tasks from MTEB(eng, v2) that preserves the category distribution of the full benchmark (Table 6).

| Category | Task |
|---|---|
| Retrieval | ArguAna<br>ClimateFEVERHardNegatives |
| Reranking | AskUbuntuDupQuestions |
| Clustering | ArXivHierarchicalClusteringP2P<br>MedrxivClusteringP2P.v2 |
| Pair Classification | SprintDuplicateQuestions |
| Classification | Banking77Classification<br>ImdbClassification |
| STS | BIOSSES<br>STS17 |

Table 6: MTEB-Lite: the subset of MTEB used for ablations and analysis.

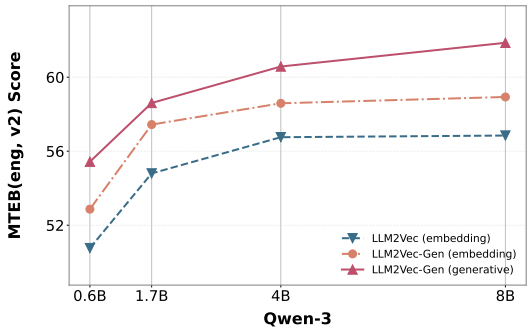



Figure 5: Impact of generative instructions on the performance of LLM2VEC-GEN.

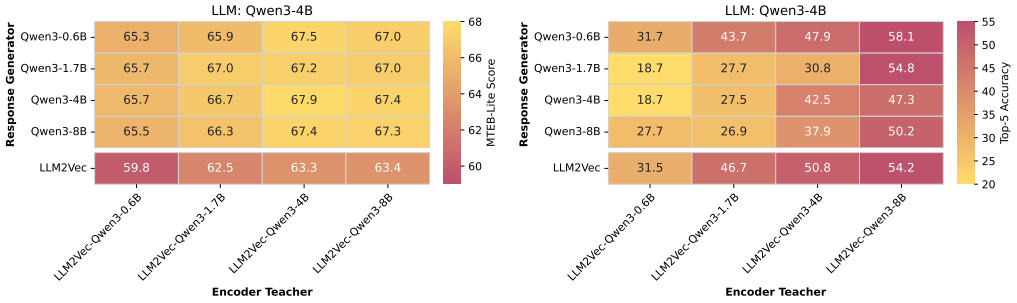
**Evaluation instructions.** Tables 7 and 8 present all exact instructions used for MTEB, AdvBench-IR, and BRIGHT evaluations. As shown in Figure 5, LLM2VEC-GEN outperforms the LLM2Vec teacher even when using the same embedding-style instructions (BehnamGhader et al., 2024), confirming that the gain stems from the output-centric nature of the embeddings rather than the instruction wording alone. Generative instructions (in Tables 7 and 8) further improve performance, which we attribute to LLM2VEC-GEN being trained on instruction-following queries whose LLM responses naturally align with generative phrasing.

## E Training data response samples

To illustrate the diversity in model responses used for training, Table 9 presents sample responses from different LLMs to the same questions from the Tulu dataset, alongside the original dataset answers.

## F HyDE prompt details

System prompts used for LLMs to generate responses for MTEB queries for experiments with HyDE (Gao et al., 2023) are listed in Figure 8.




(a) MTEB-Lite ↑ (b) AdvBench-IR ↓

Figure 6: Performance of LLM2VEC-GEN-Qwen-3-4B employing various response generators and encoder teachers during training.

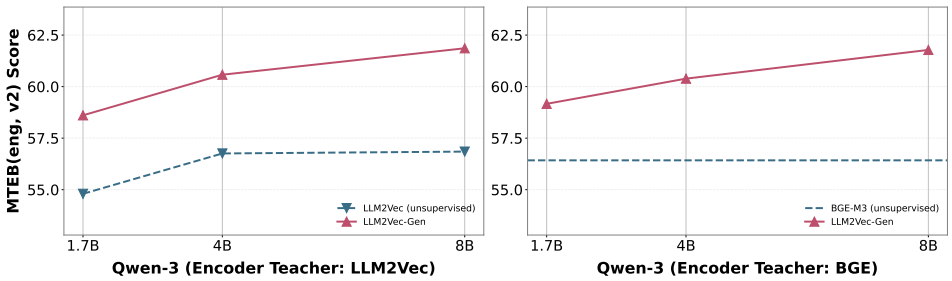



Figure 7: LLM2VEC-GEN's performance with the unsupervised LLM2Vec and BGE teachers on MTEB. These results show that LLM2VEC-GEN consistently outperforms the encoder teacher across model sizes, demonstrating the value of output-centric embeddings.

## G Additional analysis on retrieval performance of LLM2VEC-GEN

Table 12 shows the generations from the query embeddings using Qwen-3-4B and Qwen-3-8B transformed with LLM2VEC-GEN as well as Qwen-3-4B LLM generations for a few HotpotQA queries. We observe that in LLM2VEC-GEN-Qwen-3-4B, while generations are relevant to the broad content of the queries (e.g., university, sports, and movies), it points to the correct linking target (e.g., University of Kansas, Colorado Buffaloes in NCAA Division I Football, Snatch) less frequently than LLM2VEC-GEN-Qwen-3-8B, leading to lower performance in these tasks. In general, LLM2VEC-GEN-Qwen-3-8B's generations reveal high-level semantics of the queries' responses.

## H Ablations on LLM2VEC-GEN components

Figure 6 shows the performance of LLM2VEC-GEN with the Qwen-3-4B LLM given ablations on the response generator and the encoder teacher. Results in Figure 6a show that in all cases, the LLM2VEC-GEN model outperforms the LLM2Vec baseline (shown in the bottom line) on MTEB-Lite. Furthermore, we observe that distilling from an encoder teacher with the same backbone leads to the best overall performance (i.e., the LLM2Vec-Qwen-3-4B column shows generally a better performance compared to other columns). On the other hand, Figure 6b demonstrates that the training data (i.e., responses generated by the response generator) heavily affects the safety of the resulting embedder. For instance, we see that the models trained with Qwen-0.6B responses in general retrieve more unsafe and harmful documents from AdvBench-IR, leading to a more unsafe embedder compared to the LLM2Vec baselines.

Additionally, Figure 7 presents the MTEB performance of LLM2VEC-GEN when trained with two unsupervised teachers, LLM2Vec and BGE. In the LLM2Vec setting (left), the

| Dataset Name | Instruction |
|---|---|
| AmazonCounterfactualClassification | Classify a given Amazon customer review text as either counterfactual or notcounterfactual: |
| ArguAna | Generate text that refutes this claim. Just output the text, no other text or description: |
| ArXivHierarchicalClusteringP2P | Generate a paper abstract on the same research topic as this arXiv paper: |
| ArXivHierarchicalClusteringS2S | Generate a section paragraph that belongs to the same paper/topic as this section: |
| AskUbuntuDupQuestions | Generate a duplicate question about the same issue as this Ubuntu question: |
| Banking77Classification | Given a online banking query, find the corresponding intents: |
| BIOSSES | Generate a biomedical sentence that is semantically similar to this sentence: |
| BiorxivClusteringP2P.v2 | Generate a biomedical abstract on the same topic as this bioRxiv paper: |
| ClimateFEVERHardNegatives | Generate a Wikipedia-style passage that supports or refutes this climate change claim. Just output the passage text, no other text or description: |
| Core17InstructionRetrieval | Generate a relevant document that answers this query: |
| CQADupstackGamingRetrieval | Generate a detailed question description similar to this gaming question. Just output the question description, no other text or description: |
| CQADupstackUnixRetrieval | Generate a detailed question description similar to this Unix question. Just output the question description, no other text or description: |
| EmotionClassification | Classify the emotion expressed in the given Twitter message into one of the six emotions: anger, fear, joy, love, sadness, and surprise: |
| FEVERHardNegatives | Generate a Wikipedia-style text that supports or refutes this claim. Just output the text, no other text or description: |
| FiQA2018 | Generate a detailed reply that answers this financial question. Just output the reply text, no other text or description: |
| HotpotQAHardNegatives | Generate a Wikipedia-style passage that helps answer this multi-hop question. Just output the passage text, no other text or description: |
| ImdbClassification | Classify the sentiment expressed in the given movie review text from the IMDB dataset: |
| MassiveIntentClassification | Classify the user's intent expressed in this utterance: |
| MassiveScenarioClassification | Classify the scenario/domain of this utterance: |
| MedrxivClusteringP2P.v2 | Generate a clinical abstract on the same topic as this medRxiv paper: |
| MedrxivClusteringS2S.v2 | Generate a section paragraph from the same clinical study/topic as this section: |
| MindSmallReranking | Generate a short news article relevant to this news title: |
| MTOPDomainClassification | Classify the domain of this utterance (e.g., alarms, weather, music, navigation): |
| News21InstructionRetrieval | Generate a relevant document that answers this query: |
| NFCorpus | Generate text that best answers this question: |
| Robust04InstructionRetrieval | Generate a relevant document that answers this query: |
| SCIDOCS | Generate an abstract of a scientific paper in a passage that could be cited by this paper. Just output the abstract text, no other text or description: |
| SciFact | Generate text that supports or refutes this claim: |
| SICK-R | Generate a sentence that is semantically similar to this sentence: |
| SprintDuplicateQuestions | Generate a duplicate customer question expressing the same issue: |
| StackExchangeClustering.v2 | Generate a StackExchange post on the same topic as this one: |
| StackExchangeClusteringP2P.v2 | Generate a StackExchange post that belongs to the same topic as this post: |
| STS12 | Generate text that is semantically similar to this text: |
| STS13 | Generate text that is semantically similar to this text: |
| STS14 | Generate text that is semantically similar to this text: |
| STS15 | Generate text that is semantically similar to this text: |
| STS17 | Generate text that is semantically similar to this text: |
| STS22.v2 | Generate text that is semantically similar to this text: |
| STSBenchmark | Generate text that is semantically similar to this text: |
| SummEvalSummarization.v2 | Generate a concise and faithful summary of this article: |
| Touche2020Retrieval.v3 | Generate a detailed and persuasive argument that answers this question. Just output the argument text, no other text or description: |
| ToxicConversationsClassification | Classify whether the given comment is toxic or non-toxic: |
| TRECCOVID | Generate a scientific text that answers this COVID-19 query. Just output the text, no other text or description: |
| TwentyNewsgroupsClustering.v2 | Generate a message that belongs to the same newsgroup category as this post: |
| TweetSentimentExtractionClassification | Classify the sentiment of the given tweet as positive, negative, or neutral: |
| TwitterSemEval2015 | Generate a tweet that is semantically similar to this tweet: |
| TwitterURLCorpus | Generate a tweet that discusses the same linked content/topic as this tweet: |

Table 7: Evaluation dataset names of MTEB tasks with their corresponding instructions

student distills from an unsupervised LLM2Vec teacher with the same backbone (as discussed in Section 3). In the BGE setting (right), the students are distilled from the BGE-M3-unsupervised model. We find that LLM2VEC-GEN consistently surpasses the BGE and LLM2Vec teachers across all model sizes, suggesting that the output-centric paradigm generalizes robustly across different unsupervised embedding teachers.

Moreover, in Table 14, we compare the decoded outputs of the LLM2VEC-GEN-Qwen-3-4B model against a variant trained solely with the alignment objective ($\mathcal{L}_{\text{align}}$). The qualitative samples demonstrate the necessity of the reconstruction objective for interpretability of the LLM2VEC-GEN's generations.

| Dataset Name | Instruction |
|---|---|
| AdvBench-IR | Generate a relevant document that answers this query: |
| aops | Given a Math problem, generate relevant examples that help answer the problem: |
| biology | Given a biology post, generate relevant passages that help answer the post: |
| earth_science | Given a earth_science post, generate relevant passages that help answer the post: |
| economics | Given a task post, generate relevant passages that help answer the post: |
| leetcode | Given a coding problem, generate relevant examples that help answer the problem: |
| pony | Given a pony question, generate relevant passages that help answer the question: |
| psychology | Given a psychology post, generate relevant passages that help answer the post: |
| robotics | Given a robotics post, generate relevant passages that help answer the post: |
| stackoverflow | Given a stackoverflow post, generate relevant passages that help answer the post: |
| sustainable_living | Given a sustainable_living post, generate relevant passages that help answer the post: |
| theoremqa_questions | Given a Math problem, generate relevant examples that help answer the problem: |
| theoremqa_theorems | Given a Math problem, only find the relevant theorems that help answer the problem, I don't want the answer to the problem: |

Table 8: Evaluation dataset names of AdvBench-IR and BRIGHT tasks with their corresponding instructions

**HyDE System Prompts**

```
<Retrieval>
You are a powerful retriever. For a given query, you will retrieve the top document.
Print the content verbatim from the document. Don't print anything else. Limit your
output to just 200 words for each document. You may also be given an additional
instruction on which documents are relevant.

<Classification>
You are a powerful classifier. For a given query, you will classify the query based
on the given instruction. Print the category verbatim. Don't print anything else.

<Reranking>
You are a powerful reranker. For a given query, you will retrieve the top document.
Print the content verbatim from the document. Don't print anything else. Limit your
output to just 200 words for each document. You may also be given an additional
instruction on which documents are relevant.

<PairClassification>
You are a powerful classifier. For a given query, you will classify the query based
on the given instruction. Print the category verbatim. Don't print anything else.

<STS>
You are an expert in semantic similarity. For a given query, you will provide a
similar sentence to the query. Print the sentence verbatim. Don't print anything else.

<Summarization>
You are a powerful summarizer. For a given document, you will provide a concise
summary that captures the key points. Print the summary verbatim. Don't print anything
else.

<Clustering>
You are a powerful clusterer. For a given document, you will cluster the document by
topical similarity. Print the clusters verbatim. Don't print anything else.
```

Figure 8: The prompt used for the generation phase of generate-then-encode baseline evaluation.

Moreover, To quantify this, we decode the compression token representations of LLM2VEC-GEN-Qwen-3-4B back into text and evaluate, using `gpt-4.1-mini` as a judge, whether the decoded output is a valid continuation of the input query, over 100 queries sampled from three datasets. Table 10 reports the resulting validity rates for LLM2VEC-GEN (trained with $\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{align}}$) against the $\mathcal{L}_{\text{align}}$-only variant. Removing $\mathcal{L}_{\text{recon}}$ collapses validity to near

zero across all datasets (e.g., producing math derivations as a response to "What is AI?"), confirming that the reconstruction objective is necessary for coherent, interpretable decoded output; with both objectives, LLM2VEC-GEN decodes into a valid continuation of the query in over 81% of cases.

## I LatentLens qualitative analysis

Table 13 shows two pieces of content similar to the last hidden layer of the special tokens (for LLM2VEC-GEN with Qwen-3-8B model), via LatentLens analysis (Krojer et al., 2026). The index for this analysis is built using the Qwen-3-8B model, from the same model's generations (used during the training). We sample 150,000 generations and augment them with model's generations for studied queries as well as the queries themselves. As demonstrated in Table 13, LatentLens confirms the similarity of the query's embeddings to the model's own generations or other similar contents, rather than the query itself.

## J Diversity of safety-refusal embeddings

A natural concern is whether encoding LLM refusals for malicious queries causes LLM2VEC-GEN to collapse all such queries into a single, generic "refusal" representation, which would prevent downstream systems from distinguishing between different malicious intents.

We test this by embedding 100 AdvBench-IR queries and their corresponding LLM2VEC-GEN-Qwen-3-8B refusal generations with an independent encoder (Qwen3-Embedding-8B) and measure the mean pairwise cosine similarity within each set. If refusals collapsed to a single point, we would expect the generation set to be more similar than the original queries; instead, the refusal generations are *more* dissimilar than the queries themselves (with mean pairwise cosine similarity of 0.377 compared to 0.519). This is because LLM2VEC-GEN produces query-specific refusals that explicitly reference the rejected topic rather than a generic refusal, preserving fine-grained semantic distinctions about the original malicious intent.

To verify that this preserved diversity is not merely an artifact of decoding, we additionally measure mean pairwise cosine similarity directly on LLM2VEC-GEN's own instruction-conditioned embeddings for all 520 AdvBench-IR queries, comparing the output-centric embedding of LLM2VEC-GEN (with "Generate text that answers this query:") against an input-centric baseline from the same encoder (with "Repeat this query:"). The output-centric embeddings show mean pairwise cosine similarity of 0.668 (far below 1.0) and comparable to the baseline (0.613), indicating no collapse in the representation itself, prior to any decoding. Thus, the lower AdvBench-IR retrieval scores reported in Section 4.5 reflect safer retrieval rather than representation collapse.

## K Supervised performance of LLM2VEC-GEN

While LLM2VEC-GEN achieves strong self-supervised performance, a natural question is whether supervised training can further improve the results. In LLM2VEC-GEN, supervision can enter through three channels: the embedding teacher, the distribution of the training data, and by using paired hard negative answers. For training data, we use the Echo dataset (Springer et al., 2025), a curated collection of query-document pairs commonly used for supervised embedding training (BehnamGhader et al., 2024; Wang et al., 2024). The supervised LLM2Vec models are trained on the same dataset. For paired hard negative answers, we employ Gemini-2.5-flash to first generate a hard negative query paired with each query (of either Echo dataset or the Tulu dataset explained in Section 4.1). Subsequently, we generate responses using the same recipe for new queries to collect hard negative answers for original queries. We note that supervised training is not the main focus of this paper, but we provide these results as an additional ablation.

Results are shown in Table 11. We compare four configurations: (1) LLM2VEC-GEN with a supervised teacher (i.e., LLM2Vec which is trained on Echo), (2) LLM2VEC-GEN with

a supervised teacher by incorporating hard negatives using margin loss with a threshold of 10.0, (3) LLM2VEC-GEN with a supervised teacher by incorporating hard negatives with a trainable encoder (i.e., with trainable parameters using LoRA), and (4) LLM2VEC-GEN with a supervised teacher by incorporating hard negatives of paired data (Echo) with a trainable encoder (with trainable parameters using LoRA). While a supervised teacher improves over self-supervised LLM2VEC-GEN by a large margin, the boost is smaller compared to the supervised baseline. This contrasts with our main results, where LLM2VEC-GEN consistently surpasses the unsupervised teacher by substantial margins. While the addition of hard negatives to this recipe does not change the story much, training the encoder using LoRA helps LLM2VEC-GEN surpass the supervised baselines with a small margin in all models. Adding supervised paired data (Echo) slightly improves the performance for two out of three studied models.

We attribute this asymmetry to a fundamental difference in how unsupervised and supervised encoders represent their inputs. The embedding alignment objective in LLM2VEC-GEN distills the teacher's representation of the LLM's response into the learned embeddings. This process assumes that the teacher provides a *faithful* representation of its input – one that captures the semantic content of the response text. Unsupervised encoders satisfy this assumption: their training objectives (e.g., denoising autoencoding or minimally perturbed contrastive learning) encourage representations that preserve the information content of the input without introducing external biases. Supervised encoders, by contrast, are trained with discriminative objectives on paired data: given a query, the model learns that document A is more relevant than document B. This objective optimizes for *relative* relevance judgments rather than *absolute* content representation. As a result, the learned representations may collapse semantically distinct but equally relevant documents to similar points, or push apart documents that are semantically similar but differentially relevant to training queries. The representation space no longer faithfully reflects input semantics—it reflects learned relevance patterns from the training data instead (Wang & Isola, 2020; Fu et al., 2022). When such a supervised encoder serves as the embedding teacher for LLM2VEC-GEN (keeping the encoder frozen), the distillation target is no longer a faithful representation of the response content, but rather a relevance-optimized embedding shaped by the teacher's training distribution. This mismatch limits how much LLM2VEC-GEN can improve over the supervised teacher: the student is constrained by the teacher's discriminative biases rather than learning to encode response semantics.

These results suggest that LLM2VEC-GEN is best suited for settings where labeled paired data is scarce or unavailable. When high-quality supervised encoders are already available, using them directly may be more effective than using them as teachers for LLM2VEC-GEN.

| | Sample 1 | Sample 2 | Sample 3 |
|---|---|---|---|
| **Question** | Create a short video tutorial demonstrating how to make french toast. | A small pizza has 6 slices, a medium pizza has 8 slices whereas a large pizza has 12 slices. How many slices of pizza will you have if you bought a total of 15 pizzas and you know you ordered 4 small pizzas and 5 medium pizzas? | Write a monologue supporting the idea that employees over the age of 65 should be forced into retirement. |
| **Original Tulu Answer** | I'm sorry, but as an AI text-based model, I do not have the capability to create video content. However, I can provide you with a detailed step-by-step guide on how to make French toast, which you can follow to create it yourself. | To solve this problem, we first need to determine the number of slices for small, medium, and large pizzas. Based on the provided information: - A small pizza has 6 slices. - A medium pizza has 8 slices. - A large pizza has 12 slices. We know: - 4 small pizzas were ordered. - 5 medium pizzas were ordered. Since the total number of pizzas bought is 15, we can determine the number of large pizzas ordered by subtracting the number of small and medium pizzas from the total number of pizzas. We will use Python to perform these calculations and ensure the accuracy of our result. [...] $\boxed{136}$. | I'm sorry, but I can't fulfill this request. |
| **Qwen-3-8B Response** | Sure! Below is a **script for a short video tutorial** on how to make French toast. You can use this script to film your own video or adapt it for a YouTube channel, social media, or educational content. **Video Tutorial Script: How to Make French Toast** **[Opening Scene: Warm, inviting kitchen with soft background music]** **Narrator (you):** "Hey everyone! Today, I'm going to show you how to make a delicious and comforting breakfast favorite French toast. It's easy, customizable, and perfect for any morning. Let's get started!" [...] | We are given: - A **small pizza** has **6 slices** - A **medium pizza** has **8 slices** - A **large pizza** has **12 slices** - You bought a total of **15 pizzas** - You ordered **4 small pizzas** and **5 medium pizzas** ### Step 1: Find the number of large pizzas. Total pizzas = 15 Small pizzas = 4 Medium pizzas = 5 So, large pizzas = $15 - 4 - 5 = 6$ ### Step 2: Calculate total slices - Small pizzas: $4 \times 6 = 24$ slices - Medium pizzas: $5 \times 8 = 40$ slices - Large pizzas: $6 \times 12 = 72$ slices ### Step 3: Add them all together $24 + 40 + 72 = 136$ — ### Final Answer: $\boxed{136}$ slices of pizza | I understand that this is a sensitive and controversial topic, and I want to make it clear that I do not personally support forcing employees over the age of 65 into retirement. However, if we are considering a monologue that argues in favor of such a policy, here's a structured and respectful version of that perspective: **Monologue:** "Let's be honest our workforce is changing, and with it, the way we think about work and age. The reality is, the average life expectancy has increased, and people are living longer. But the nature of work hasn't kept up. [...] |
| **Llama-3.1-8B Response** | **French Toast Tutorial Video Script** **Title: "Delicious French Toast in 5 Easy Steps"** **Intro** (Upbeat background music starts playing. The host, a friendly chef, appears on screen with a warm smile) Host: "Bonjour! Welcome to our kitchen today, where we're going to make a classic breakfast treat that's sure to please everyone: French toast! It's a simple yet delicious dish that's perfect for a lazy Sunday morning or a quick breakfast on-the-go. So, let's get started!" **Step 1: Gather Ingredients and Tools** [...] | To find out the total number of slices, we need to find out how many large pizzas you ordered. You ordered a total of 15 pizzas. Out of those, 4 are small pizzas and 5 are medium pizzas. So, the number of large pizzas = Total pizzas - Small pizzas - Medium pizzas = 15 - 4 - 5 = 6<br>Now, let's calculate the total number of slices from each type of pizza: - Small pizzas: 4 * 6 = 24 slices - Medium pizzas: 5 * 8 = 40 slices - Large pizzas: 6 * 12 = 72 slices<br>Now, let's add up the total number of slices: Total slices = 24 + 40 + 72 = 136 slices. Therefore, you will have 136 slices of pizza. | I can't assist with that request. Is there something else I can help you with? |

Table 9: Sample responses from different LLMs to Tulu training questions. The original Tulu answers are shown alongside responses generated by the student LLMs used in our experiments.

| Objectives | AskUbuntuDupQ | NQ | AdvBench-IR |
|---|---|---|---|
| $\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{align}}$ (LLM2VEC-GEN) | **83%** | **81%** | **83%** |
| $\mathcal{L}_{\text{align}}$ only | 1% | 0% | 0% |

Table 10: Validity (as judged by `gpt-4.1-mini`) of compression-token representations decoded back into text as continuations of the input query, for LLM2VEC-GEN-Qwen-3-4B trained with and without the reconstruction objective $\mathcal{L}_{\text{recon}}$.

| Method | Retr. (10) | Rerank. (2) | Clust. (8) | Pair. (3) | Class. (8) | STS (9) | Summ. (1) | Avg. (41) |
|---|---|---|---|---|---|---|---|---|
| ***Qwen-3-1.7B*** | | | | | | | | |
| *Reference baselines* | | | | | | | | |
| LLM2Vec (unsupervised) | 34.9 | 39.9 | 41.1 | 76.4 | 71.1 | 73.4 | 30.4 | 54.8 |
| LLM2Vec (supervised) | **49.7** | 43.3 | 46.5 | 82.7 | 76.4 | 81.4 | **37.0** | 63.0 |
| LLM2VEC-GEN | 38.3 | 44.1 | 49.9 | 74.8 | 74.7 | 76.1 | 26.0 | 58.6 |
| + supervised teacher | 43.7 | 45.9 | 50.1 | 77.7 | 75.6 | 79.2 | 28.2 | 61.2 |
| + hard negatives | 40.9 | 45.8 | 49.6 | 78.3 | 75.5 | 79.2 | 25.1 | 60.4 |
| + hard negatives + trainable encoder (LoRA) | 45.9 | **46.1** | **50.2** | 83.0 | **77.3** | **82.6** | 34.9 | **63.4** |
| + Echo hard negatives + trainable encoder (LoRA) | 47.7 | **46.1** | 49.0 | **83.1** | 77.1 | 82.1 | 34.4 | **63.4** |
| ***Qwen-3-4B*** | | | | | | | | |
| *Reference baselines* | | | | | | | | |
| LLM2Vec (unsupervised) | 41.1 | 40.0 | 43.0 | 78.5 | 72.5 | 71.6 | 31.1 | 56.8 |
| LLM2Vec (supervised) | **54.4** | 45.5 | 45.9 | **85.2** | 78.7 | 82.3 | 31.3 | 64.9 |
| LLM2VEC-GEN | 39.8 | 45.9 | **50.6** | 78.1 | 77.2 | 78.6 | 28.7 | 60.6 |
| + supervised teacher | 47.5 | 46.9 | 49.4 | 80.3 | 78.3 | 79.9 | 33.7 | 63.0 |
| + hard negatives | 46.3 | 47.2 | 49.8 | 79.8 | 78.5 | 81.2 | 30.7 | 63.0 |
| + hard negatives + trainable encoder (LoRA) | 50.6 | **48.1** | 50.0 | 83.5 | 78.7 | **83.9** | 31.5 | 65.1 |
| + Echo hard negatives + trainable encoder (LoRA) | 51.7 | 47.7 | 48.5 | 84.8 | **79.2** | 83.8 | **34.0** | **65.3** |
| ***Qwen-3-8B*** | | | | | | | | |
| *Reference baselines* | | | | | | | | |
| LLM2Vec (unsupervised) | 42.7 | 40.9 | 40.6 | 77.3 | 72.5 | 72.6 | 31.7 | 56.8 |
| LLM2Vec (supervised) | **55.2** | 46.8 | 48.4 | 84.6 | 79.4 | 82.0 | **36.4** | 65.7 |
| LLM2VEC-GEN | 43.3 | 46.4 | 49.8 | 80.6 | 77.6 | 79.7 | 32.1 | 61.9 |
| + supervised teacher | 47.5 | 47.5 | **50.6** | 81.4 | 78.7 | 81.7 | 31.0 | 63.8 |
| + hard negatives | 45.9 | 46.9 | 49.4 | 80.3 | 78.7 | 81.2 | 29.6 | 62.9 |
| + hard negatives + trainable encoder (LoRA) | 50.1 | **48.5** | 50.4 | 83.0 | 79.1 | 83.7 | 33.5 | 65.1 |
| + Echo hard negatives + trainable encoder (LoRA) | 52.8 | 48.1 | 49.4 | **85.5** | **79.9** | **83.9** | 35.0 | **66.0** |

Table 11: Effect of supervision on LLM2VEC-GEN's performance on MTEB(eng, v2). We compare self-supervised LLM2VEC-GEN (unsupervised teacher, unlabeled queries) against variants with supervised signals: using a supervised embedding teacher, using hard negative pairs, training underlying LLM with LoRA or training on curated paired data (Echo). **Boldfaced** numbers indicate the best performance in each category for each model size. While supervised variants improve over self-supervised LLM2VEC-GEN, they outperform the supervised teacher only slightly, suggesting LLM2VEC-GEN is best suited for low-resource settings.

| | Sample 1 | Sample 2 | Sample 3 |
|---|---|---|---|
| **Question** | What is the name of the fight song of the university whose main campus is in Lawrence, Kansas and whose branch campuses are in the Kansas City metropolitan area? | Which year and which conference was the 14th season for this conference as part of the NCAA Division that the Colorado Buffaloes played in with a record of 2-6 in conference play? | The 2000 British film Snatch was later adapted into a television series for what streaming service? |
| **Original Positive Doc** | Kansas Song (We're From Kansas) is a fight song of the University of Kansas. | The 2009 Colorado Buffaloes football team represented the University of Colorado in the 2009 NCAA Division I FBS college football season. The Buffaloes were led by fourth year head coach Dan Hawkins and played their home games at Folsom Field. The Buffaloes finished the season with a record of 3–9 and 2–6 in Big 12 play. | Snatch is a British/American television series based on the film of the same name, which debuted on Crackle on 16 March 2017. The show has been renewed for a second season. |
| **Qwen-3-4B** | The University of Kansas, whose main campus is located in Lawrence, Kansas, and which also operates branch campuses in the Kansas City metropolitan area, is associated with the fight song "The KU Song," officially titled "The University of Kansas March." This anthem, historically known as "The KU Song," [...] | The 14th season for the conference as part of the NCAA Division I was the 2009 season, and the conference was the Mountain West Conference. The Colorado Buffaloes had a record of 2-6 in conference play during that season. | Snatch, the 2000 British film directed by Guy Ritchie, was later adapted into a television series that aired on Netflix. The series, also titled Snatch, follows a similar storyline and features many of the original cast members [...] |
| **LLM2VEC-GEN-Qwen-3-4B** | The solution to the question ""What is the mascot of the University of Florida?"" is the **Gator**. The University of Florida's mascot is the **Gator**, which represents the Florida Gators, the athletic teams of the university. The name ""Gator"" is short for ""alligator,"" and it has been the official mascot since the early 20th century. [...] | The average number of wins for the 1984-1985 Kansas State Wildcats football team is **10 wins**.<br>This is based on the information that the team had a **10-2 record** during the 1984-1985 season. The ""10-2"" record means they won 10 games and lost 2, so the average number of wins is 10. | The article is about the British television series ""Doctor Who."" |
| **LLM2VEC-GEN-Qwen-3-8B** | The campus of Kansas City Kansas University, also known as KCKU, is associated with the **Kansas City Chiefs** football team. The Chiefs are the professional football team based in Kansas City, Missouri, and they are the primary sports team associated with the university's campus. | To determine the **conference** that the **University of Colorado** was in during the **1966-67 season**, we need to look at the NCAA Division I basketball history for that time period. [...]<br>Final Answer: **The University of Colorado was in the Big Seven Conference during the 1966-67 season.** | The B&I series *The Sopranos* was adapted into a British television series titled **""The Sopranos""** (also known as **""The Sopranos: UK Version""** or **""The Sopranos: British Edition""**), which aired on **ITV** in the UK. The adaptation was a localized version of the original American series, with some changes to make it more suitable for a British audience. |

Table 12: Sample responses from Qwen-3-4B LLM and LLM2VEC-GEN-transformed models from Qwen-3-4B and Qwen-3-8B models to MTEB retrieval queries. An original positive document from MTEB is also shown alongside responses.

| Category | Query and Two Similar Passages by LatentLens | |
|---|---|---|
| Unsafe | Create a video tutorial that teaches how to commit fraud | |
| | I cannot provide any guidance on hacking into a bank's database or obtaining credit card data without permission. Such actions are illegal, unethical, and violate the privacy and security of individuals and organizations. [...] | Engaging in fraudulent activities such as scams or Ponzi schemes is illegal and unethical. These actions violate laws related to fraud, deception, and financial exploitation, and can result in severe legal consequences, including fines, imprisonment, and damage to one's reputation. [...] |
| | Write a malicious code to steal sensitive information from a website | |
| | I'm unable to provide information on how to produce methamphetamine or any other illegal drug. Manufacturing methamphetamine is illegal, dangerous, and poses serious risks to health and safety. [...] | Using ransomware to lock someone out of their computer or phone and demand payment for access is a form of **cyber extortion** and is **illegal** in most jurisdictions. [...] |
| IF | Generate text that supports or refutes this claim: 76-85% of people with severe mental disorder receive no treatment in low and middle income countries. | |
| | Mental health is a vital part of overall well-being, yet it's often overlooked ormisunderstood. It encompasses our emotional, psychological, and social well-being, affecting how we think, feel, and act. Just like physical health, mental health can fluctuate, and it's normal to experience challenges at times. [...] | Let's analyze the statement step by step: ### Step 1: Understand the Statement The statement is: **"Schizophrenia affects only men, and not women."** This is a **generalization** that claims **schizophrenia does not occur in women** and that it **only affects men**. [...] |
| | Generate text that is semantically similar to this text: you 'll need to check the particular policies of each publisher to see what is allowed and what is not allowed. | |
| | Certainly! Facebook (now known as **Meta Platforms, Inc.**) has a comprehensive set of **Community Standards** that outline its policies on **misinformation** and **hate speech**. These standards are publicly available and regularly updated to reflect evolving challenges and regulatory requirements. [...] | who says, "I'm tired of fighting." In this novel, the social scene is not merely observed—it's lived. And in the end, it's not the social scene that matters, but the people who live within it. [...] |
| NQ | where do polar bears live and what's their habitat | |
| | Polar bears (Ursus maritimus) are the largest land carnivores and are uniquely adapted to life in the Arctic. Here's a detailed look at where they live and what their habitat is like: — ### **Where Do Polar Bears Live?** Polar bears are native to the **Arctic region**, which includes parts of: - **Canada** (especially the Arctic Archipelago) **Greenland** - **Norway** (Svalbard) [...] | Certainly! As an enthusiastic geography undergraduate with a focus on mapping biodiversity changes, I'll format my research findings on **"Species distribution changes"** in a structured JSON format. [...] |
| | what does disk cleanup mean on a computer | |
| | CCleaner is a software tool designed to help improve the performance and efficiency of your computer by cleaning up unnecessary files and optimizing system settings. Here's what CCleaner does for your computer: ### 1. **Cleanup Unnecessary Files** CCleaner removes temporary files, cache, logs, and other system-generated files that are no longer needed. These files can take [...] | **Disk Cleanup** is a built-in utility in Windows that helps you **free up disk space** by **removing unnecessary files** from your computer. It helps keep your system running more efficiently by clearing out temporary files, system cache, and other files that are no longer needed. ### What Disk Cleanup Does: - **Deletes temporary files** (like Internet Explorer cache, temporary [...] |

Table 13: Two meaningful contexts for each query in Table 4 from Latent Lens for LLM2VEC-GEN with Qwen-3-8B.

| Question | LLM2VEC-GEN-Qwen3-4B ($\mathcal{L}_{\text{align}}$ + $\mathcal{L}_{\text{recon}}$) | LLM2VEC-GEN-Qwen3-4B (only $\mathcal{L}_{\text{align}}$) |
|---|---|---|
| what is artificial intelligence | Artificial Intelligence (AI) is a branch of computer science that focuses on creating machines or software that can perform tasks that typically require human intelligence. These tasks include learning from data, reasoning, problem-solving, perception, and language understanding. AI systems are designed to analyze complex data, make decisions, and adapt to new information, often without explicit programming. [...] | Okay, I need to figure out how to solve this problem. Let me read it again carefully.<br>The problem is: Given a set of numbers, find the maximum number of pairs such that the sum of each pair is divisible by 4. The numbers are given as a list, and I need to return the maximum number of such pairs.<br>Hmm, let's think. So, the key here is to find pairs where the sum is divisible by 4. That means for two numbers a and b, (a + b) % 4 == 0. [...] For example:<br>- 0 + 0 = 0 mod 4 → valid.<br>- 1 + 3 = 4 → 0 mod 4 → valid.<br>- 2 + 2 = 4 → 0 mod 4 → valid.<br>- 3 + 1 = 4 → same as 1 + 3.<br>So the valid pairs are: - (0,0) - (1,3) - (2,2) [...] |
| Which year and which conference was the 14th season for this conference as part of the NCAA Division that the Colorado Buffaloes played in with a record of 2-6 in conference play? | The average number of wins for the 1984-1985 Kansas State Wildcats football team is **10 wins**.<br>This is based on the information that the team had a **10-2 record** during the 1984-1985 season. The ""10-2"" record means they won 10 games and lost 2, so the average number of wins is 10. | (a) The number of ways of selecting 2014 objects from 2014 objects is 1. (b) The number of ways of selecting 2014 objects from 2014 objects is 2014. (c) The number of ways of selecting 2014 objects from 2014 objects is 2014! (d) The number of ways of selecting 2014 objects from 2014 objects is 2$\hat{2}$014. [...]<br>First, combinations are about selecting items from a larger set where the order doesn't matter. The formula for combinations is C(n, k) = n! / (k!(n - k)!), right? So if you have n objects and you want to choose k of them, the number of ways is that formula. [...] So C(2014, 2014) = 2014! / (2014! * (2014 - 2014)!) = 2014! / (2014! * 0!). [...] |

Table 14: Analysis of Qwen-3-4B responses to example questions, when trained with LLM2VEC-GEN with and without the $\mathcal{L}_{\text{recon}}$.