CtrlShift: Steering Language Models for Dense Quotation Retrieval with Dynamic Prompts

Chuang Liang and **Yanqiu Shao*** and **Wei Li**
School of Information Science, Beijing Language and Culture University, China
smetase@outlook.com, yqshao163@163.com, liweitj47@blcu.edu.cn

Abstract

Quotation recommendation is an inherently asymmetric retrieval task, where the intended meaning of a quote often diverges from surface expressions, creating significant semantic shifts. Combined with minimal lexical overlap, this poses a core challenge for classic dense retrievers, which struggle to capture non-literal and rhetorical alignments. To bridge this semantic gap, we propose introducing controllable signals to guide the model's attention toward abstract, context-relevant concepts. We propose CTRLSHIFT, a framework that leverages a Variational Autoencoder (VAE) to capture latent associations between context and quotation, which is used to derive context-aware control signals to modulate semantic focus and support bidirectional alignment and rhetorical intent modeling. Experiments show that our method consistently outperforms baselines on the quotation recommendation task and can be effectively transfered to the general purposed benchmark. Further, CtrlShift integrates seamlessly with general-purpose generative models without additional fine-tuning, and provides satisfactory interpretability by generating textual explaination to uncover the model's focus on abstract, citation-aligned semantics.

1 Introduction

Quotation recommendation, the task of retrieving classical excerpts to enrich modern literature (Tan et al., 2015), serves as a powerful tool for enhancing rhetorical expression. However, this task poses a significant challenge for standard dense retrieval (DR) models, revealing fundamental limitations in their design. As our preliminary experiments in Appendix Table 7 show, even state-of-the-art embedding models perform poorly, underscoring the need for a different retrieval paradigm.

This performance gap arises from the intrinsic properties of the task. Quotation recommendation is inherently *asymmetric* (Liao et al., 2024); modern contexts and classical quotes differ starkly in style, abstraction, and vocabulary (Qi et al., 2022). As illustrated in Figure 1 (right), relevance depends less on lexical overlap and more on **functional alignment**. Quotations often rely on **metaphor or imagery**, introducing a gap between surface form and intended meaning—what we term a semantic shift. Tellingly, interaction-heavy models like ColBERT (Khattab and Zaharia, 2020), which rely on fine-grained token similarity, perform even worse (see Appendix Table 8), suggesting that over-reliance on surface matching is counterproductive. This need for functional alignment challenges traditional retrieval systems designed for semantic similarity (Thakur et al., 2021).

The reliance of dense retrievers on surface-level lexical signals is well-documented; they often fail to capture salient keywords (Karpukhin et al., 2020; Chen et al., 2021) and tend to prioritize superficial overlaps over factual or functional relevance (Fayyaz et al., 2025). As a result, they struggle to model the kinds of semantic shifts and abstract alignments required for effective quotation recommendation. While commonly used (Wu and Cao, 2024; Metzler et al., 2021), pseudo-query generation is unstable and unreliable in open-ended citation tasks (Abe et al., 2025).

Importantly, recent embedding models, especially those based on decoder-only LLMs (Chen et al., 2024; Muennighoff et al., 2024; Wang et al., 2024a), exhibit emergent capabilities (Wei et al., 2022) that arise from scale and representation learning. These models inherently possess the capacity to capture abstract reasoning and contextual nuance, offering a bottom-up mechanism for modeling semantic drift and latent alignment.

We propose a modular soft control mechanism to dynamically steer embedding gener-

*Corresponding Author

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2192–2204
December 20-24, 2025 ©2025 Association for Computational Linguistics

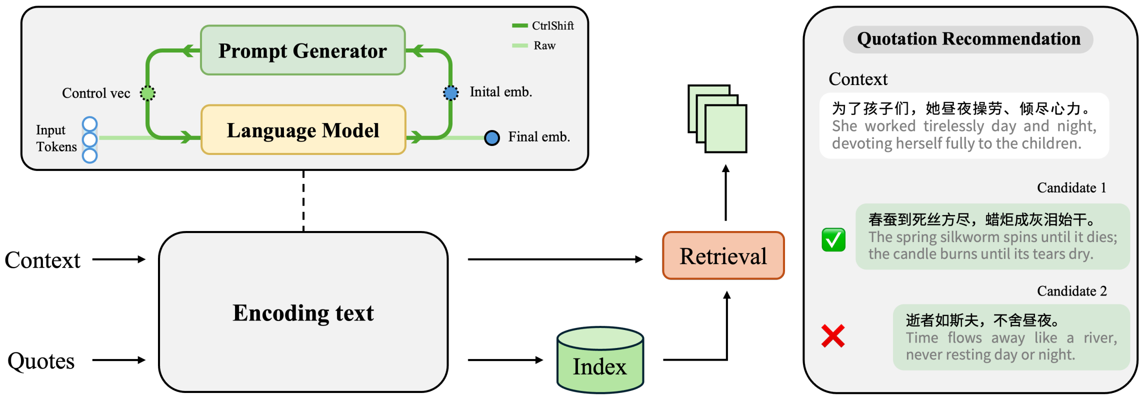

Figure 1: An overview of our CTRLSHIFT framework. **Left**: The main pipeline, featuring a shared encoder and a two-stage process. An initial embedding is passed through an external prompt generator to produce a dynamic control vector, which is injected into a frozen language model to yield a refined, context-aware representation. **Right**: Illustrative examples demonstrating that effective quotation matching relies on deeper functional alignment rather than mere surface-level lexical overlap.

ation—shifting focus from surface-level token overlap to abstract, functional semantics. As shown in Figure 1 (right), this enables the model to move beyond superficial matches (e.g., "day and night") and instead align with contextually relevant concepts (e.g., "selfless dedication"), even in the absence of lexical overlap.

To this end, we introduce **CTRLSHIFT**, a lightweight framework that equips frozen language models with dynamic, context-aware embedding capabilities. As illustrated in Figure 1 left, CTRLSHIFT follows a two-stage process: an initial embedding is produced, then a lightweight control module—implemented as a VAE—derives a context-sensitive control vector. This vector is injected back into the LLM to yield refined embeddings aligned with abstract semantics. The entire framework is trained end-to-end with a self-supervised objective.

We conducted extensive experiments demonstrating that CTRLSHIFT improves performance across multiple languages and generalizes well to MS MARCO. This is significant because direct fine-tuning on this saturated benchmark often degrades performance by disrupting the model's pre-trained knowledge (Pande et al., 2025). Our method avoids this pitfall by adapting the model without altering its weights. Furthermore, it enables general-purpose LLMs to produce competitive embeddings without task-specific tuning, and supports interpretability via decoding of abstract control signals.

Our contributions are as follows:

- We present **CTRLSHIFT**, a lightweight control framework that explores a novel form of model self-refinement. It enables fine-grained semantic modulation of frozen language models by using a VAE to learn latent, context-aware concepts for functional alignment.
- We demonstrate that CtrlShift achieves consistent and significant performance gains on the specialized quotation recommendation task, and generalizes robustly to the general-purpose MS MARCO benchmark.
- We show that CTRLSHIFT enables effective retrieval with general-purpose decoder-only language models, without task-specific fine-tuning, and inherently supports interpretability by decoding control vectors into abstract citation-related concepts, leveraging the generative capabilities of LLMs.

2 Related Work

Dense Retrieval Dense retrieval (DR) encodes queries and documents into a shared embedding space to support efficient retrieval beyond lexical matching. The field has evolved from bi-encoders using contrastive finetuning with negative sampling (Karpukhin et al., 2020; Xiong et al., 2021) to modern models pretrained at scale like E5 (Wang et al., 2022), GTE (Li et al., 2023), and BGE (Chen et al., 2024). To overcome bi-encoder limitations, their capabilities are often enhanced by distilling knowledge from more powerful but inefficient cross-encoders (Rosa et al., 2022; Qu et al., 2021; Ren et al., 2021a; Zhang et al., 2021; Ren et al., 2021b).

The advent of Large Language Models (LLMs)

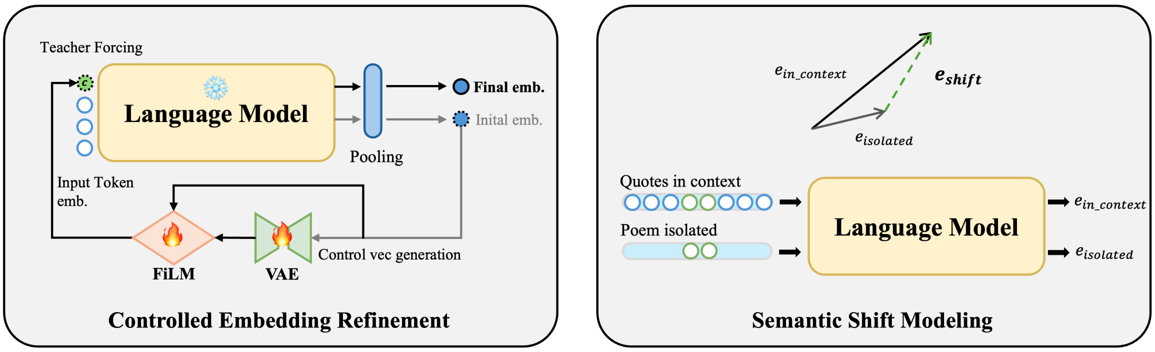

Figure 2: The core mechanisms of CTRLSHIFT. **(Left)** Controlled Embedding Refinement: An initial embedding is modulated by a FiLM-controlled VAE to produce a control vector, which is then injected into the frozen language model to generate the refined final embedding. **(Right)** Semantic Shift Modeling: The VAE learns to model the semantic shift (e_{shift}), defined as the difference between the in-context ($e_{in\_context}$) and isolated ($e_{isolated}$) item embeddings.

has spurred new embedding models, from decoder-only architectures (Liu et al., 2024; Wang et al., 2024b; Lee et al., 2024a,b) to specialists created by fine-tuning generative models like Gemini-embedding (Lee et al., 2025) and Qwen3-embedding (Zhang et al., 2025) on synthetic data (Wang et al., 2024a). However, these models act as **static encoders**, unable to leverage their instruction-following ability for dynamic contextual adaptation—a core limitation our work addresses.

Prompting for Retrieval Prompting improves dense retrieval in a parameter-efficient way. Most prior prompting methods in retrieval rely on static strategies (Peng et al., 2025; Lee et al., 2022; Ma et al., 2022), including instruction-based prompting with synthetic data (Dai et al., 2022; Asai et al., 2022; Su et al., 2022; Wang et al., 2024a). These global approaches overlook input-specific semantics. While dynamic prompting has been explored for reranking (Wu et al., 2024), we introduce the first dynamic control mechanism for dense retrieval.

Quotation Recommendation Quotation recommendation has evolved from a learning-to-rank task with hand-crafted features (Tan et al., 2015) to early neural models (LSTMs/CNNs) (Tan et al., 2016, 2018; Ahn et al., 2016). Research has since improved semantic alignment using structured knowledge (Xu et al., 2022; Liu et al., 2021), established benchmarks (Qi et al., 2022), and extended the task to dialogue and generation (Lee et al., 2016; Wang et al., 2021; Xiao et al., 2024).

We are the first to frame this task from a modern dense retrieval perspective, with **CTRLSHIFT** designed to capture deep, context-dependent relevance beyond surface similarity.

3 Approach

As shown in Figure 1 (left), CTRLSHIFT reformulates dense retrieval as a two-stage process: generating a general-purpose embedding followed by context-aware refinement. The name CTRLSHIFT reflects our core idea—using a dynamically generated control(Ctrl) vector to capture the semantic shift of text in context. The main pipeline, Controlled Embedding Refinement (Figure 2, left), is guided by Semantic Shift Modeling (Figure 2, right), which provides auxiliary supervision for the control vector.

3.1 Problem Formulation

Let C be an input context and $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ be a corpus of N source poems. The objective is to retrieve the specific poem $P_j \in \mathcal{P}$ that is functionally and semantically aligned with the context C.

We formulate this as a dense retrieval task, aiming to learn an embedding function $f(\cdot)$ that maps both contexts and poems into a shared semantic space $\mathbb{R}^d$. For a given context C, the model is trained to ensure that its embedding $f(C)$ is closer to that of the source poem $f(P_j)$ than to any non-source poem $f(P_i)$ $(i \neq j)$, under a similarity metric $sim(\cdot, \cdot)$. Following the standard dense retrieval pipeline, all poems in the corpus P are encoded offline via $f(\cdot)$ to construct an embedding index. At inference time, C is encoded into a query vector and matched against the index to retrieve top-ranked candidates.

3.2 Controlled Embedding Refinement

Our refinement process enables model self-adaptation via an external control mechanism. By generating dynamic control vectors, it steers a frozen language model toward functional, context-aware semantics suitable for asymmetric retrieval. As shown in Figure 2 (left), this process is parameter-efficient and leaves the base LLM untouched.

We begin by generating an initial embedding einit, which is passed through a lightweight Variational Autoencoder (VAE) (Kingma et al., 2013) to produce a latent variable $\mathbf{z}$ capturing the abstract "citation concept." A Feature-wise Linear Modulation (FiLM) layer (Perez et al., 2018) conditions einit on $\mathbf{z}$, and a ControlHead transforms $\mathbf{z}$ into a dynamic control vector $\mathbf{c}$:

$$\mathbf{c} = \gamma(\mathbf{z}) \odot \mathbf{e}_{\text{init}} + \text{ControlHead}(\mathbf{z}) \quad (1)$$

where $\gamma(\cdot)$ and ControlHead$(\cdot)$ are MLPs that generate scaling and shifting parameters, respectively. This operation preserves the richness of $\mathbf{e}_{\text{init}}$ while aligning it with the structured abstraction in $\mathbf{z}$, enabling precise semantic refinement without modifying the language model.

3.3 Semantic Shift Modeling

While our end-to-end retrieval objective implicitly encourages the model to understand contextual meaning, we introduce **Semantic Shift Modeling** as an auxiliary objective to make this process more explicit and robust. This approach is conceptually grounded in the distributional hypothesis (Firth, 1957) and the additive properties of word embeddings (Mikolov et al., 2013). Inspired by relational embedding models that model relations as translations in vector space (Bordes et al., 2013; Wang et al., 2014), we explicitly model the semantic shift a poem undergoes.

As shown in Figure 2 (right), the shift vector $\mathbf{e}_{\text{shift}}$ is defined as:

$$\mathbf{e}_{\text{shift}} = \mathbf{e}_{\text{in\_context}} - \mathbf{e}_{\text{isolated}} \quad (2)$$

This vector is intended to capture the contextual transformation of the poem's semantics. To guide this process, we train the control vector $\mathbf{c}$ to approximate the semantic shift vector $\mathbf{e}_{\text{shift}}$ using an auxiliary loss (see Section 3.4). This additional supervision encourages the control module to model nuanced, context-dependent meaning, which we hypothesize to be beneficial for achieving better functional alignment.

3.4 Training Objectives

CTRLSHIFT is trained end-to-end using multiple objectives that jointly encourage structured latent representations and controllable, context-sensitive semantics.

VAE Regularization. To ensure the latent variable z being able to capture rich and generalizable semantic features, we adopt a variational autoencoding setup. A KL-divergence loss encourages the posterior distribution to remain close to a standard Gaussian prior:

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left(q_\phi(z \mid \mathbf{e}_{\text{init}}) \parallel \mathcal{N}(0, \mathbf{I})\right) \quad (3)$$

Retrieval Loss. To align the learned embeddings with downstream retrieval objectives, we adopt an InfoNCE loss (Oord et al., 2018). Given a context embedding e_P and its corresponding positive poem embedding e_P^+, along with a set of negative poem embeddings $e_{P_i}^-$, the loss is:

$$\mathcal{L}_{\text{retrieval}} = -\log p^* \quad (4)$$

$$p^* = \frac{\exp\left(\frac{\mathbf{q}^\top \mathbf{p} - m}{\tau}\right)}{\exp\left(\frac{\mathbf{q}^\top \mathbf{p} - m}{\tau}\right) + \sum_{\mathbf{q}^- \in \mathcal{N}(\mathbf{q})} \exp\left(\frac{(\mathbf{q}^-)^\top \mathbf{p}}{\tau}\right)} \quad (5)$$

where $sim(\cdot, \cdot)$ is cosine similarity and τ is a temperature hyperparameter. Negatives are sampled from within the batch.

Semantic Shift Prediction Loss. To further guide latent learning, we introduce an auxiliary reconstruction objective that explicitly supervises semantic transformations. A decoder conditioned on $\mathbf{z}$ predicts a shift vector $\hat{\mathbf{e}}_{\mathbf{shift}}$, trained to match a reference shift embedding $\mathbf{e}_{\mathbf{shift}}$ derived from the context-poem pair:

$$\mathcal{L}_{\text{shift}} = \|\hat{\mathbf{e}}_{\text{shift}} - \mathbf{e}_{\text{shift}}\|_2^2 \quad (6)$$

This loss anchors the latent space to interpretable transformations, encouraging z to encode controllable semantic variations. As shown in our ablations, incorporating this shift supervision leads to more structured and effective representations.

Backbone	Method	English			Modern Chinese			Traditional Chinese		
		MRR	nDCG	R@10	MRR	nDCG	R@10	MRR	nDCG	R@10
BGE-M3 (Encoder-only)	Raw	0.0936	0.1039	16.28	0.0950	0.1061	17.11	0.0700	0.0848	13.26
	Pseudo Query	0.1088	0.1213	19.05	0.1106	0.1251	20.31	0.0805	0.0895	14.42
	Data Aug	0.1334	0.1480	22.33	0.1178	0.1337	21.62	0.1023	0.1148	18.04
	CTRLSHIFT	**0.4456**	0.4655	**59.78**	**0.4230**	**0.4567**	**58.59**	**0.3441**	**0.3752**	**50.11**
	P-tuing v2	0.4379	**0.4698**	59.68	0.3286	0.3637	50.13	0.3195	0.3529	48.80
Qwen3-E-4B (Decoder-only)	Raw	0.1818	0.2032	30.29	0.1362	0.1544	24.37	0.1451	0.1641	25.63
	Pseudo Query	0.1088	0.1213	24.81	0.0626	0.0683	11.06	0.0577	0.0633	10.70
	Data Aug	0.1587	0.1776	26.82	0.1094	0.1227	19.59	0.1195	0.1361	21.77
	CTRLSHIFT	**0.5876**	**0.6243**	**75.87**	**0.4796**	**0.5232**	**68.20**	**0.4382**	**0.4834**	**65.02**
	P-tuing v2	0.5416	0.5860	74.74	0.3632	0.4066	57.15	0.3767	0.4232	60.06
Qwen3-E-0.6B (Decoder-only)	Raw	0.0470	0.0502	8.21	0.0434	0.0467	7.79	0.0375	0.0400	6.62
	Pseudo Query	0.0363	0.0394	6.72	0.0346	0.0374	6.55	0.0266	0.0277	4.88
	Data Aug	0.0465	0.0506	8.25	0.0401	0.0436	7.50	0.0291	0.0321	5.76
	CTRLSHIFT	**0.4974**	**0.5439**	**67.02**	**0.4040**	**0.4434**	**59.28**	**0.3665**	**0.4065**	**56.12**
	P-tuing v2	0.4742	0.5122	65.74	0.3005	0.3299	45.30	0.3376	0.3735	51.65

Table 1: Quotation retrieval performance (MRR, nDCG, Recall@10) across diverse backbones and languages. CTRLSHIFT consistently outperforms both the P-tuning v2 baseline and the "Raw" baseline baselines while being significantly more efficient (one input token vs. 64 per-layer tokens in P-tuing v2).

Part	Train	Val	Test	Total
English	101,171/6,008	12,771/6,108	12,771/6,108	126,713/6,108
mChinese	32,472/2,904	4,185/3,004	4,185/3,004	40,842/3,004
tChinese	93,031/4,338	11,753/4,438	11,753/4,438	116,537/4,438

Table 2: Statistics of the QUOTER dataset. Each entry m/n denotes m context–quote pairs involving n unique quotes.

Overall Loss. The total training objective $\mathcal{L}_{\text{total}}$ is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{retrieval}} + \lambda_2 \mathcal{L}_{\text{kl}} + \lambda_3 \mathcal{L}_{\text{shift}}, \quad (7)$$

where λ_1, λ_2, and λ_3 are hyperparameters that control the relative importance of each loss component.

4 Experiments

We evaluate CTRLSHIFT on the QuoteR benchmark (Qi et al., 2022) (Table 2), a large-scale dataset designed to test retrieval under metaphorical shifts, low lexical overlap, and domain-specific semantics, providing a robust testbed for our method.

We treat quote recommendation as a single-stage dense retrieval task, where the input is a passage and the goal is to retrieve the most semantically aligned quote. Models are evaluated using Recall@10, nDCG, and MRR.

4.1 Computational Resources

All experiments are conducted on a single NVIDIA A800 80GB GPU using the official PyTorch 2.5.1 container. Further implementation details and the code repository are provided in Section A.2.

4.2 Baselines

We compare CTRLSHIFT against two primary baselines that use a unified dual-encoder architecture with parameter-efficient tuning: (1) a Raw baseline that directly uses the pooled rep- resentations from the pretrained models, and (2) P-tuning v2, implemented via DPTDR (Ma et al., 2022), a strong prompt-based dense retrieval approach. We also include the results from the original QuoteR paper (Qi et al., 2022) as a key historical benchmark. It is important to note the significant methodological differences between our approach and the QuoteR baseline. The QuoteR model is an independent dual-encoder that undergoes multi-stage, full fine-tuning. In contrast, both CTRLSHIFT and P-tuning v2 employ a unified dual-encoder (i.e., a shared backbone) and use lightweight prompt tuning, keeping the base model frozen. Furthermore, the original QuoteR task assumes a specific insertion point for the quote, whereas our setup addresses the more general task of retrieving a relevant quote for an entire passage.

4.3 Main Results

We present the main results in Table 1, which reveals a clear and consistent pattern: CTRLSHIFT substantially and consistently improves

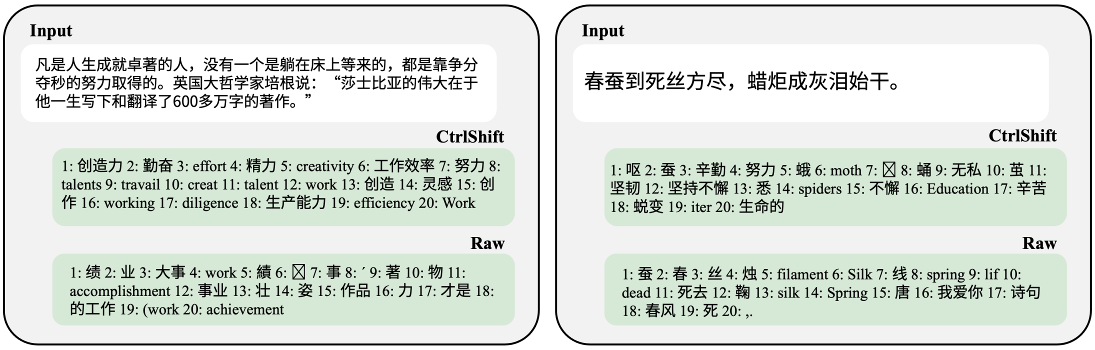

Figure 3: Qualitative analysis of the semantic space. By decoding embeddings, we show that CTRLSHIFT steers the model's focus from surface-level keywords (Raw) to abstract, functional concepts for both a modern context (left) and a classical poem (right).

Model Variant	Recall@10	MRR	nDCG
Full CTRLSHIFT	56.12	0.4065	0.3665
w/o Shift	52.09	0.3410	0.3628
w/o FiLM	54.75	0.3576	0.371
w/o VAE	8.25	0.0371	0.0825

Table 3: Ablation of key components. Both shift modeling and VAE are critical.

performance across all tested backbones (BGE-M3, Qwen3 series) and language datasets (English, Modern Chinese, and Traditional Chinese). Our framework consistently outperforms both the unmodified base models and the strong prompt-based P-tuning v2 baseline. We also experimented with two auxiliary enhancements—pseudo query generation (Pseudo Query) and data augmentation (Data Aug) via document explanation. However, these enhancements provided only limited and inconsistent overall benefit across models.

Crucially, our method achieves these gains with remarkable efficiency. While P-tuning v2 injects 64 virtual tokens per layer, CtrlShift adds only a single token to the input, making it significantly more lightweight. Despite this efficiency, our method achieves performance that is competitive with the fully fine-tuned QuoteR. This demonstrates that our parameter-efficient, external control mechanism can match a much heavier, multistage, full fine-tuning approach, highlighting the power and efficiency of our method.

Due to GPU memory limitations, P-tuning v2 required a smaller batch size and gradient accumulation for stable training. The detailed training and inference efficiency comparison for both models is provided in Appendix A.3. Furthermore, P-tuning v2 lacked native support for backbones utilizing grouped query attention (GQA, Ainslie et al., 2023), necessitating implementation-level adjustments for models like Qwen3.

4.4 Ablation Studies

We conduct ablation studies on the Traditional Chinese dataset using Qwen3-embedding-0.6B as the backbone to assess the impact of key components. As shown in Table 3, each architectural element contributes meaningfully to overall performance.

Removing the semantic shift prediction loss ("w/o Shift") led to a noticeable drop in retrieval performance, underscoring the importance of modeling contextual transformation explicitly. Disabling the FiLM layer ("w/o FiLM") similarly degraded performance, indicating its role in effectively modulating the control signal. Most notably, replacing the VAE with a simple two-layer MLP bottleneck ("w/o VAE") resulted in the most severe performance degradation. This highlights the limitations of a deterministic bottleneck and confirms the VAE's effectiveness in learning a structured latent space crucial for dynamic control.

4.5 Qualitative Analysis: Interpreting the Semantic Space

To analyze the effect of CTRLSHIFT, we decode the final embedding vectors via the model's decoder head (`lm_head`), using the top predicted tokens as proxies for semantic focus.

As illustrated in Figure 3 (with English translations in Appendix A.4), CTRLSHIFT shifts representations from surface-level co-occurrences (e.g., "achievement", "work") to more abstract, meaning-driving concepts (e.g., "creativity", "diligence"). For classical texts, it redirects outputs from literal tokens (e.g., "silkworm", "spring") toward deeper thematic concepts such as "selfless" and "perseverance". These results suggest that CtrlShift guides the model toward functional semantics over lexical overlap. To our knowledge, this is the first work to leverage decoder-only LLM-based embeddings for interpretability in dense retrieval, offering simple yet effective semantic insight.

To substantiate this functional semantic shift, we performed K-Means clustering (K=50) on Raw and CtrlShift embeddings, and decoded the cluster centroids for semantic labeling (Due to space limitations, the full cluster tables are available in our code repository (URL in Appendix A.2).

The analysis reveals a marked contrast: Raw embeddings often yield opaque or metadata-centric cluster centroids (e.g., author names or genre tags), reflecting an organization heavily influenced by surface co-occurrence. In contrast, CtrlShift embeddings consistently produce centroids that capture higher-level thematic concepts. Clusters that were previously lexical now shift towards abstract qualities like "Diligence" or functional roles such as "Literary Creation."

This analysis demonstrates that CtrlShift effectively reorganizes the semantic space around functional roles and abstract themes, yielding highly interpretable embeddings and better functional alignment.

Effect of Control Target. To examine whether explicitly modeling *semantic shift* improves control effectiveness, we compare our default target embedding ($\mathbf{e}_{\text{shift}}$) with two alternatives: the embedding of the full poem within context ($\mathbf{e}_{\text{poem-in-context}}$) and that of the isolated poem alone ($\mathbf{e}_{\text{poem-isolated}}$). As shown in Table 4, $\mathbf{e}_{\text{shift}}$ consistently yields the best performance across all metrics. In contrast, using the full poem or isolated poem as the target leads to substantial drop in retrieval quality, likely due to semantic ambiguity or overfitting to surface features. All models show consistent gains when integrated with our framework, suggesting its potential generality and applicability.

4.6 Analysis of Implementation Choices

We evaluate design choices for pooling and control vector injection using the Traditional Chinese dataset and Qwen3-0.6B backbone (Figure 4).

Control Target	Recall@10	MRR	nDCG
$\mathbf{e}_{\text{shift}}$ **(default)**	56.12	0.3665	0.4065
$\mathbf{e}_{\text{poem-in-context}}$	53.62	0.3441	0.3752
$\mathbf{e}_{\text{poem-isolated}}$	54.76	0.3512	0.3809

Table 4: Comparison of control targets. Modeling the semantic shift vector is most effective.

Table 5: Retrieval performance of a specialized embedding model vs. a general-purpose LLM, with and without CtrlShift.

Model	Method	R@10	MRR	nDCG
Embed-0.6b	Raw	6.62	0.0375	0.0400
	CtrlShift	**56.12**	**0.3665**	**0.4065**
LLM-0.6b	Raw	1.04	0.0082	0.0073
	CtrlShift	**53.61**	**0.3538**	**0.3905**

Pooling Strategies As shown in Figure 4a, While standard approaches like Mean Pooling and Last Token Pooling are common, they can be suboptimal; mean pooling may dilute important semantic signals, while last-token pooling may not capture the full context of a sequence. Our results confirm that Latent Attention (Lee et al., 2024a), which uses a learnable query to perform task-adaptive aggregation of token-level hidden states, achieves the best performance. This highlights the benefit of a more expressive and flexible pooling mechanism for our task.

Control Vector Injection We also compare four strategies for injecting the control vector c into the frozen LLM (Figure 4).b). The simplest method, Add, which merely perturbs the input embeddings, yields the poorest results, suggesting a weak conditioning effect. Prepend and Append, which insert c as pseudo-tokens, perform better but are still significantly outperformed by our main approach. The Attach strategy proves decisively superior. By treating c as a virtual token injected directly via the model's `past_key_values` cache, it allows the LLM to strongly and directly condition its final representation on our control signal without any architectural modifications. This result indicates that direct autoregressive conditioning is a more effective mechanism for semantic modulation than simple input sequence manipulation.

4.7 Unifying Generative and Embedding Models

A key motivation for our work is to explore the potential of using a single, general-purpose gen-

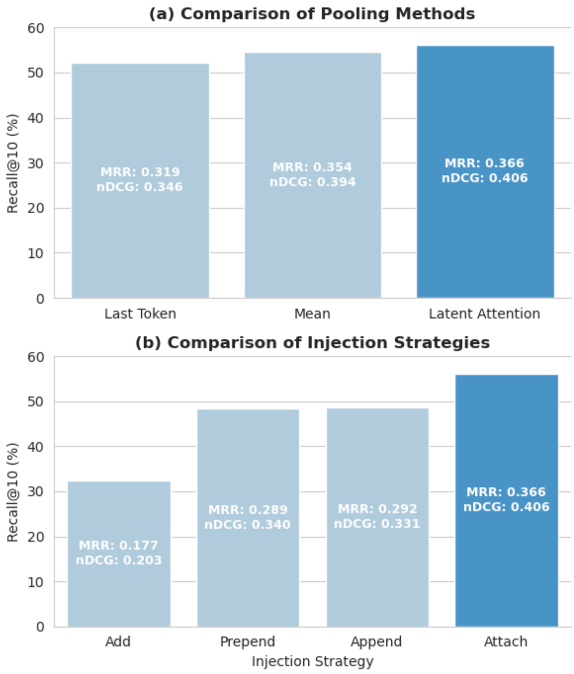

Figure 4: Ablation study on pooling methods (a) and control vector injection strategies (b). Results on the Traditional Chinese dataset with the Qwen3-0.6B backbone show that Latent Attention and the Attach strategy yield the best performance.

erative LLM for both text generation and high-quality text embedding. To this end, we conducted an experiment comparing the retrieval performance of a specialized embedding model (Qwen3-embedding-0.6b) with a general-purpose generative model (Qwen3-0.6b) of a similar scale.

As shown in Table 5, the raw generative model (Qwen3-0.6b) performs poorly on the retrieval task, achieving an nDCG of only 0.0073. This is expected, as it was not trained for discriminative embedding tasks. However, when augmented with our CTRLSHIFT framework, its performance dramatically improves to an nDCG of 0.3905.

Remarkably, this result is nearly identical to the performance of the specialized Qwen3-embedding-0.6b model equipped with CTRLSHIFT (0.4065 nDCG). This demonstrates that our lightweight control mechanism can effectively steer a general-purpose generative model to produce embeddings that are competitive with state-of-the-art specialized models, without requiring any fine-tuning of the base model's weights. This finding highlights a promising path toward unifying text generation and representation learning within a single, versatile architecture.

4.8 Generalization to Other Benchmarks

Table 6: Performance on the MS MARCO passage ranking dev set. CTRLSHIFT maintains the strong performance of the base models, avoiding the performance degradation often seen with fine-tuning on this benchmark.

Model	Recall@10	MRR	nDCG
BGE-M3	53.26	0.4668	0.4813
Qwen3-embedding-0.6b	53.47	0.4688	0.4836

To test generalization beyond quotation recommendation, we evaluated CTRLSHIFT on the MS MARCO passage ranking benchmark (Bajaj et al., 2016). We report results on the development set (as the test set is unavailable), restricting retrieval to the labeled passages due to memory constraints.

As shown in Table 6, applying CTRLSHIFT preserves the high performance of strong base models like BGE-M3 and Qwen3-embedding-0.6b on this general-domain task. The two models perform comparably, as expected given their similar scale.

This result is notable given recent findings that fine-tuning strong sentence transformers on MS MARCO can degrade performance by disrupting the semantic structure built during large-scale pre-training (Pande et al., 2025). In contrast, CTRLSHIFT leaves the base model unchanged, adapting its representations externally via a lightweight control signal. This preserves pre-trained knowledge while improving task-specific alignment—a particularly beneficial property on saturated benchmarks.

5 Conclusion

We introduce CTRLSHIFT, a lightweight framework that steers a frozen language model via dynamic control vectors to capture functional, context-aware semantics for asymmetric retrieval tasks. Our experiments show this approach significantly improves performance on quotation recommendation and generalizes robustly to standard benchmarks like MS MARCO, notably avoiding the performance degradation common to fine-tuning on saturated benchmarks. Furthermore, by enabling general-purpose generative models to produce embeddings competitive with specialized retrieval systems, our work highlights a promising path toward unifying representation learning and generation through dynamic semantic control.

Ethical Considerations

This work focuses on retrieval-based writing assistance, a relatively low-risk application domain. All evaluations are conducted on publicly available datasets (e.g., QuoteR), promoting transparency and reproducibility.

However, since our framework builds on large language models, it may inherit biases or stereotypical associations from the underlying models, potentially leading to inappropriate outputs. We do not recommend deployment in sensitive contexts where such risks could cause harm. While some interpretability analyzes are included, further work is needed to ensure transparency and robustness. Our method is lightweight in terms of parameter updates, though we do not quantify its environmental impact.

Limitations

While our method enables self-refinement by guiding a frozen model via external control, it relies on supervised context-quote pairs to train the control mechanism, which may limit applicability in low-resource settings. Our exploration of this mechanism is also preliminary; though results are promising, experiments are limited in scale and model diversity.

In addition, while we introduce a dataset with human-verified citation rationales to support rationale-aware evaluation, its current coverage is narrow. Future work should expand this dataset and further analyze how control vectors reshape semantic space and capture transferable latent concepts.

Acknowledgements

This research project is supported by the National Natural Science Foundation of China (No.62306045), the Humanities and Social Science Project of the Ministry of Education (24YJAZH119), Science Foundation of Beijing Language and Culture University (supported by "the Fundamental Research Funds for the Central Universities") (18ZDJ03).

References

Kenya Abe, Kunihiro Takeoka, Makoto P Kato, and Masafumi Oyamada. 2025. Llm-based query expansion fails for unfamiliar and ambiguous queries. *arXiv preprint arXiv:2505.12694*.

Yeonchan Ahn, Hanbit Lee, Heesik Jeon, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation for dialogs and writings. In *CBRecSys@ RecSys*, pages 39–42.

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. 2025. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. *arXiv preprint arXiv:2503.05037*.

John Rupert Firth. 1957. *Papers in Linguistics 1934–1951*. Oxford University Press.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via late interaction with bert. In *SIGIR*.

Diederik P Kingma, Max Welling, et al. 2013. Auto-encoding variational bayes.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024a. Nv-embed: Improved techniques for training llms as generalist embedding models.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 957–960.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. 2024b. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*.

Zhuyun Lee et al. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning.

Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. 2024. D2LLM: Decomposed and distilled large language models for semantic search. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14798–14814, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoyang Liu, Yi Zhang, et al. 2021. Metaphor-aware poem generation with conceptual mappings. In *ACL*.

Zheng Liu, Chaofan Li, Shitao Xiao, Yingxia Shao, and Defu Lian. 2024. Llama2Vec: Unsupervised adaptation of large language models for dense retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3490–3500, Bangkok, Thailand. Association for Computational Linguistics.

Xiaodong Ma et al. 2022. Dptdr: Deep prompt tuning for dense passage retrieval. *arXiv preprint arXiv:2208.11503*.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: making domain experts out of dilettantes. *SIGIR Forum*, 55(1).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Manu Pande, Shahil Kumar, and Anay Yatin Damle. 2025. When fine-tuning fails: Lessons from ms marco passage ranking. *arXiv preprint arXiv:2506.18535*.

Zhiyuan Peng, Xuyang Wu, Qifan Wang, and Yi Fang. 2025. Soft prompt tuning for augmenting dense retrieval with large language models. *Knowledge-Based Systems*, 309:112758.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Fanchao Qi, Yanhui Yang, Jing Yi, Zhili Cheng, Zhiyuan Liu, and Maosong Sun. 2022. QuoteR: A benchmark of quote recommendation for writing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 336–348, Dublin, Ireland. Association for Computational Linguistics.

Yifan Qu, Yuxing Liu, Lei Yang, et al. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL*.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. Pair: Leveraging passage-centric similarity relation for improving dense passage retrieval. *arXiv preprint arXiv:2108.06027*.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *arXiv preprint arXiv:2110.07367*.

Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Jiwei Tan, Xiaojun Wan, Hui Liu, and Jianguo Xiao. 2018. Quoterec: Toward quote recommendation for writing. *ACM Transactions on Information Systems (TOIS)*, 36(3):1–36.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Learning to recommend quotes for writing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 65–74.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Lingzhi Wang, Jing Li, Xingshan Zeng, Haisong Zhang, and Kam-Fai Wong. 2021. Continuity of topic, interaction, and query: Learning to quote in online conversations. *arXiv preprint arXiv:2106.09896*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Mingrui Wu and Sheng Cao. 2024. Llm-augmented retrieval: enhancing retrieval models through language models and doc-level embedding. *arXiv preprint arXiv:2404.05825*.

Xuyang Wu, Zhiyuan Peng, Krishna Sravanthi Sai Rajanala, Hsin-Tai Wu, and Yi Fang. 2024. Passage-specific prompt tuning for passage reranking in question answering with large language models. *arXiv preprint arXiv:2405.20654*.

Jin Xiao, Bowei Zhang, Qianyu He, Jiaqing Liang, Feng Wei, Jinglei Chen, Zujie Liang, Deqing Yang, and Yanghua Xiao. 2024. Quill: Quotation generation enhancement of large language models. *arXiv preprint arXiv:2411.03675*.

Lee Xiong, Chenyan Wu, Donald Metzler, et al. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR*.

Ming Xu, Yichao Zhang, et al. 2022. Knowledge-enhanced classical chinese poetry recommendation via graph neural networks. In *COLING*.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*.

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. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Appendix

A.1 Embedding Model Performance

Table 7: Performance of strong embedding models on classical poetry citation retrieval. Standard models struggle to capture the required functional and asymmetric relevance.

Model	Recall@10	MRR	nDCG
BGE-M3	13.26	0.0700	0.0848
GTE-Qwen2-7B	**24.29**	**0.1284**	**0.1556**
GTE-Qwen2-1.5B	20.68	0.1060	0.1298
E5-large	12.65	0.0651	0.0796

Table 8: ColBERT underperforms on poetic citation tasks, suggesting that fine-grained token interactions alone are insufficient for capturing semantic resonance.

Model	Recall@10	MRR	nDCG
BGE-M3	13.26	0.0700	0.0848
BGE-M3 (ColBERT)	12.41	0.0631	0.0774

A.2 Training Environment

All experiments are conducted on a single NVIDIA A800 80GB GPU using the official

PyTorch 2.5.1 container. We employ the PyTorch Lightning framework with mixed-precision training (bfloat16) to improve computational efficiency.

Optimization is performed using AdamW with default weight decay. The learning rate is dynamically adjusted via a ReduceLROnPlateau scheduler, which reduces it by a factor of 0.5 if the validation performance plateaus for more than 3 consecutive epochs. Early stopping is applied with a patience of 5 epochs based on validation retrieval metrics. These strategies improve convergence and generalization across model variants. Our code is available at https://github.com/sMetase/CtrlShift.

Table 9: Training and model hyperparameters for CTRLSHIFT.

Hyperparameter	Value
vae_latent_dim	128
vae_hidden_dim	512
free_nats	0.8
loss_retrieval_temp	0.035
batch_size	256
accumulate_grad_batches	1
epoch	25
k_recall	10
loss_weight_formulas.loss_kl	0.02 * progress
loss_weights.loss_pred	1.0
loss_weights.loss_retrieval	1.0
lr	0.002
lr_decay_factor	0.5
lr_scheduler_type	plateau

Table 9 summarizes the hyperparameters used for training CTRLSHIFT. For the P-tuning v2 baseline, we adopt a different tuning configuration better suited for prompt-based methods, as detailed in Table 10.

Table 10: Additional hyperparameters specific to P-tuning v2.

Hyperparameter	Value
batch_size	128
accumulate_grad_batches	2
num_virtual_token	64

A.3 Training and Inference Efficiency

A.4 English translation of the qualitative analysis

Table 11: Training Resource Usage

Resource	CtrlShift	P-tuning v2
Training Time	4.235 hr	3.798 hr
GPU Memory	~20895 MB	~75131 MB
Batch Size	256	128

Table 12: Inference Resource Usage

Resource	CtrlShift	P-tuning v2
Inference Batch Size	512	512
GPU Memory	65362 MB	40125 MB
Time Encoding	24.3 s	22.4 s

Table 13: **English translation of the qualitative analysis.**

Input Text	Method	Top Decoded Tokens
"Anyone who has achieved great things in life did not get them by waiting in bed, but through tireless effort. The great British philosopher Bacon said: 'Shakespeare's greatness lies in the fact that he wrote and translated over 6 million words in his lifetime.'"	CTRLSHIFT	1: creativity, 2: diligence, 3: effort, 4: energy, 5: creativity, 6: work efficiency, 7: effort, 8: talents, 9: travail, 10: creat, 11: talent, 12: work, 13: creation, 14: inspiration, 15: creative work, 16: working, 17: diligence, 18: productivity, 19: efficiency, 20: work
	Raw	1: achievement, 2: deed, 3: great event, 4: work, 5: achievement, 6: –, 7: matter, 8: –, 9: notable, 10: thing, 11: accomplishment, 12: career, 13: grand, 14: form, 15: work of art, 16: effort, 17: is, 18: work, 19: work, 20: achievement
"The spring silkworm spins until it dies; the candle burns until its tears dry."	CTRLSHIFT	1: exert, 2: silkworm, 3: hardworking, 4: effort, 5: moth, 6: moth, 7: –, 8: pupa, 9: selfless, 10: cocoon, 11: tenacious, 12: perseverance, 13: all, 14: spiders, 15: unremitting, 16: education, 17: hardship, 18: transformation, 19: iter, 20: of life
	Raw	1: silkworm, 2: spring, 3: silk, 4: candle, 5: filament, 6: silk, 7: thread, 8: spring, 9: life, 10: dead, 11: die, 12: bow, 13: silk, 14: spring, 15: Tang, 16: I love you, 17: verse, 18: spring breeze, 19: die, 20: –