AN LLM HIDES SEVERAL SEMANTIC EXPERTS: PROBING TASK-SPECIFIC ATTENTION VALUES FOR TEXT EMBEDDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Different text embedding tasks require different notions of semantic similarity, motivating representations that adapt to the task. In this paper, we investigate whether its attention head modules already contain task-specific semantic signals. Specifically, we introduce a training-free framework that probes and ranks individual modules using augmented text pairs or downstream reference data, then constructs compact task-specific embeddings from their attention values. Two complementary augmentations enable selection using unlabeled data; task instructions can also select and combine dataset-specific experts. Probing is performed offline and routing is decided once per task, while each text requires only one frozen-model forward pass. Across 14 MTEB tasks, eight selected modules yield 1024-dimensional embeddings that exceed the strongest prompt-based ensemble by 4.66 and 5.08 average points on Llama3-8B and Qwen3-8B, respectively. Each text requires one forward pass; the ensemble uses eight. Dynamic routing achieves the highest mean at this dimension on both models. On a trained LLM2Vec embedder, selected modules surpass its native 4096-dimensional representation with either 2048 or 4096 dimensions. These results show that offline task-specific probing can turn existing dense models into flexible, compact embedders.

1 INTRODUCTION

Text embedding is one of the core technologies in natural language processing. It underpins large-scale text mining tasks such as text clustering and dense retrieval, and serves as the fundamental foundation for many LLM-augmented frameworks such as retrieval-augmented generation (RAG) (Lewis et al., 2020) and agentic memory systems (Xu et al., 2026).

Given the text $x \sim X$, the text embedding task aims to learn a function F that projects text into a d-dimensional vector $\mathbf{z} \sim Z \subset \mathbb{R}^d$. In the early days, the task-specific text embedder F_t is trained for the fixed downstream tasks $t \sim T$, i.e., $F_t : X_t \to Z_t$ (Gao et al., 2021; Karpukhin et al., 2020). As data from various tasks is accumulated and integrated, the research attempts to train a general text embedding model, i.e., $F : X \to Z$, that can be used for multiple types of tasks (Muennighoff, 2022; Neelakantan et al., 2022; Li et al., 2023; Chen et al., 2024). However, as research has progressed, it has become clear that the different properties of the embedding space are required for different tasks (Hämmerl et al., 2023; Li et al., 2024; Zhang et al., 2025a), making general-purpose text embeddings difficult to obtain. Consequently, research has gradually shifted toward developing a general-purpose embedder $G : T \times X \to Z$ instead.

To produce different embeddings of the same text for different tasks, one approach is to map task t to a text instruction I_t. Concatenating the instruction with the text then yields a task-specific embedding, $G(t, x) = F(I_t \bigoplus x)$ (Su et al., 2023; Peng et al., 2024; Zhang et al., 2025c). A second approach maps task t to a task-specific function F_t. Existing expert-based approaches use mixture-of-experts (MoE) LLMs or train task-specific embedding experts to realise $G(t, x) = F_t(x)$ (Li & Zhou, 2025; Calvo et al., 2025; Kachuee & Sharifkhani, 2026).

We take the expert-based view to components already present in a dense LLM. The central question is whether individual attention head modules encode semantic signals that favor different embedding tasks. If such specialization is present, a task can be mapped to selected modules, and the attention

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

values extracted from them can form its text embedding. This connects the model's internal component structure to task-conditioned representation design.

This idea draws on two lines of research. Analyses of Transformer components show that their functions and representations vary across the model (Dar et al., 2023; Zhang et al., 2025b). Separately, training-free embedding studies examine which internal states of an LLM best support semantic comparison (Li & Zhou, 2025; Skean et al., 2025; Zhang et al., 2026). In particular, Zhang et al. (2026) show that attention values can outperform hidden states, attention queries, attention keys, and feed-forward activations as embeddings. They select intermediate layers using downstream datasets and average their attention values to obtain a general-purpose representation. These findings motivate a closer look at the modules that constitute each layer's value representation.

Building on this finding, we develop **training-free** attention-value embeddings through three connected components: (1) probing data that expose useful module rankings, (2) module-level combinations that control representation size, and (3) task-specific routing that links those combinations to target tasks. The first component identifies candidate semantic experts; the other two determine how their values are used for encoding.

For the *probing data*, we use **text augmentations** and a **contrastive loss** to score modules from unlabeled corpora. CROP and ICT provide complementary views of semantic matching: the former pairs related short spans, while the latter relates a sentence to its longer context. Their rankings resemble those obtained from downstream reference datasets, and their selected modules achieve comparable embedding performance. Rankings are also reproducible within a fixed probing-data distribution. We examine full-ranking correlation and Top-8 overlap separately to show how changes in the probing budget affect ordering and the selected set. This makes augmentation-based scores useful when the target task has limited labeled data for module selection.

For the *embedding source*, we combine the values of selected **attention head modules**. Layer-wise pooling has already shown that attention values are informative, but selecting complete layers fixes the granularity of the representation. Each module in the models studied here contributes 128 dimensions, so module-level selection gives finer control over the embedding budget. The resulting combinations can draw semantic information from different depths of the model. Since the space of possible combinations is large, we rank modules individually and concatenate the highest-ranked values. This simple construction yields a 256-dimensional representation that rivals the best layer-wise attention-value embedding.

For *task routing*, module rankings show **task-specificity** across probing-text forms, languages, and domains. A single ordering therefore cannot capture every target task equally well. We treat the top-ranked modules under each probing source as a semantic expert and connect target tasks to appropriate experts. Static routing chooses sources by task category; dynamic routing uses the similarity of task instructions to select and combine dataset-specific experts. Both routes reuse offline scores and the frozen model, while allowing the selected representation to vary with the task. This makes task specialization possible at the module level during inference.

In the training-free setting, 1024-dimensional embeddings selected from augmented data outperform the strongest prompt-based ensemble (4096 dimensions and eight forward passes) by over five average-score points on Qwen3-8B and 4.66 on Llama3-8B across 14 MTEB tasks, while approaching downstream-probed attention-value embeddings. On trained LLM2Vec-Llama3-Sup (BehnamGhader et al., 2024), selected-module embeddings exceed its native 4096-dimensional embedding even at 2048 dimensions.

2 Background

2.1 Prompt-Based Training-Free Text Embeddings

Prompt-based methods extract text embeddings from frozen LLMs using natural-language templates. PromptEOL (Jiang et al., 2024) requests a one-word summary and uses the hidden state at the final prompt position as the sentence embedding. Contrastive prompting (CP) extends PromptEOL with an auxiliary prompt to steer intermediate representations at inference time (Cheng et al., 2025). Echo Embeddings (EE) (Springer et al., 2025) repeats the input in a prompt and pools representations from its second occurrence, allowing each token to attend to the complete first copy. MetaEOL (Lei

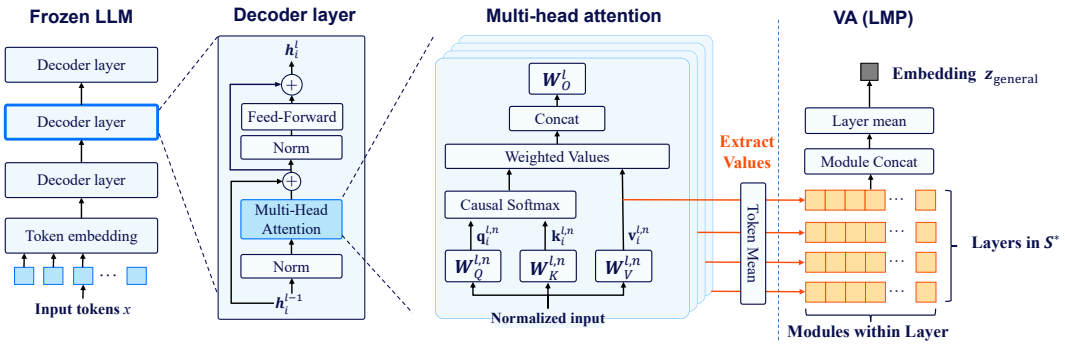


Figure 1: Transformer decoder architecture and layer-wise attention-value embeddings. Values are extracted before attention-weighted mixing; VA (LMP) pools complete layer values over tokens and selected layers $\mathcal{S}^*$.

et al., 2024) averages embeddings from eight meta-task prompts that elicit complementary semantic information, requiring eight forward passes per example.

2.2 Module Composition of Transformer Decoder-based LLMs

For a text of N tokens, an L-layer decoder LLM has hidden width d and h attention heads per layer, with head dimension $d_h = d/h$. For token i, head n in layer l projects the normalized hidden state into query, key, and value vectors:

$$[\mathbf{q}_i^{l,n}; \mathbf{k}_i^{l,n}; \mathbf{v}_i^{l,n}] = [\mathbf{W}_Q^{l,n}; \mathbf{W}_K^{l,n}; \mathbf{W}_V^{l,n}] \, \mathrm{Norm}(\mathbf{h}_i^{l-1}) \in \mathbb{R}^{3d_h}. \tag{1}$$

Here each projection is in $\mathbb{R}^{d_h \times d}$, and semicolons denote vertical concatenation. Attention mixes values across tokens before output projection and the feed-forward residual update; we extract value vectors before this mixing (Figure 1).

Grouped query attention (GQA) (Ainslie et al., 2023) reduces KV-cache storage (Pope et al., 2023) by sharing keys and values among groups of g query heads. This leaves h/g distinct KV modules per layer. Henceforth, n indexes these distinct groups, and the complete layer value is

$$\mathbf{v}_i^l = \mathrm{Concat}(\mathbf{v}_i^{l,1}, \ldots, \mathbf{v}_i^{l,h/g}) \in \mathbb{R}^{d_g}, \qquad d_g = d_h(h/g) = d/g. \tag{2}$$

The concatenated queries $\mathbf{q}_i^l$ have dimension d, while compact keys $\mathbf{k}_i^l$ have dimension d_g; $g = 1$ recovers standard multi-head attention.

2.3 Layer-wise Attention Values as Text Embeddings

Text embeddings are typically pooled with the hidden states from the final Transformer layer. Some pioneering works (Li & Li, 2024; Skean et al., 2025) find that text embeddings pooled by the middle-layer hidden states are superior to those pooled by the last hidden states. Zhang et al. (2026) further explores ways to obtain better text embedding in the intermediate layers and empirically finds that attention-value-based embedding $\mathbf{v}^l = \frac{1}{N}\sum_{i=1}^N \mathbf{v}_i^l$ outperforms the embedding pooled with query ($\mathbf{q}^l = \frac{1}{N}\sum_{i=1}^N \mathbf{q}_i^l$), key ($\mathbf{k}^l = \frac{1}{N}\sum_{i=1}^N \mathbf{k}_i^l$), and hidden states ($\mathbf{h}^l = \frac{1}{N}\sum_{i=1}^N \mathbf{h}_i^l$) in almost all layers. In practice, the authors select several downstream task datasets for evaluation and choose a set of layers that perform well overall on these downstream tasks to form the layer set $\mathcal{S}^*$. We denote this approach by VA (LMP), where LMP stands for Layer-wise Mean Pooling. As shown on the right of Figure 1, the text embedding is obtained by mean pooling over $\mathcal{S}^*$:

$$\mathbf{z}_{\text{general}} = \frac{1}{|\mathcal{S}^*|}\sum_{l\in\mathcal{S}^*} \mathbf{v}^l = \frac{1}{N|\mathcal{S}^*|}\sum_{l\in\mathcal{S}^*}\sum_{i=1}^N \mathbf{v}_i^l \in \mathbb{R}^{d_g}. \tag{3}$$

Layer-wise pooling establishes attention values as an embedding source. We next score their modules for specific tasks and assemble compact representations from selected values.

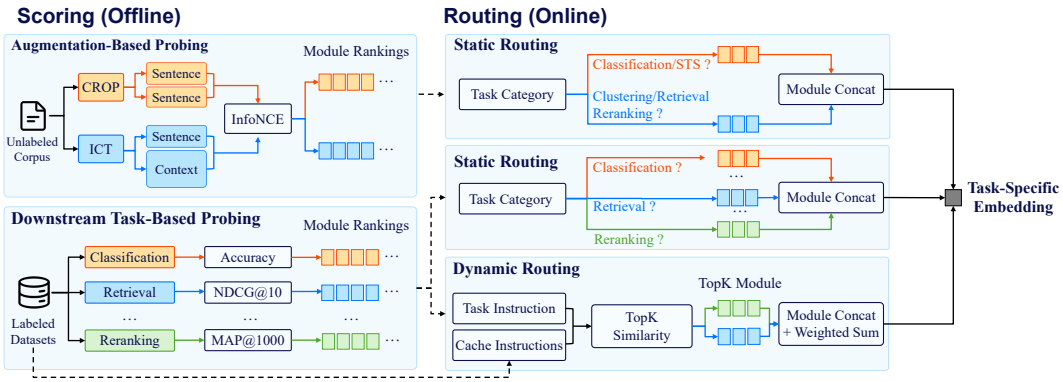


Figure 2: Overview of module scoring and task routing. CROP/ICT and TaskEval rank attention head modules offline; static and dynamic routes construct task-specific embeddings from their values.

2.4 Mixture-of-Experts Text Embedding Models

There have been some attempts in current research to use MOE LLMs to obtain text embeddings. For example, sparse MoE text embedders train routed experts (Nussbaum & Duderstadt, 2025), while MoTE introduces task experts for multi-task embedding (Calvo et al., 2025). Existing MoE LLMs have also been examined as training-free embedding models (Li & Zhou, 2025). These studies motivate task-conditioned expert choice. Here, offline scores define experts from attention head modules already present in a frozen dense LLM, and task-level routing chooses among them.

3 Attention Values as Task-Specific Embeddings

Our framework separates offline module scoring from task-level routing (Figure 2). We rank attention head modules using augmented or downstream data, then route target tasks to the resulting experts. Indices j and t denote probing sources and target tasks, respectively.

3.1 Probing Data and Module Scoring

Let $\mathcal{M} = \{(l, n) : 1 \le l \le L, 1 \le n \le h/g\}$ denote the distinct attention head modules. For an input text x, each module provides a value embedding by mean pooling over its N_x valid body tokens, excluding padding and terminal tokens by default:

$$\mathbf{z}^{l,n}(x) = \frac{1}{N_x} \sum_{i=1}^{N_x} \mathbf{v}_i^{l,n}(x) \in \mathbb{R}^{d_h}. \tag{4}$$

The $\binom{Lh/g}{k}$ possible subsets make direct combination search costly. We therefore score each module individually on D_j using one of two approaches.

Augmentation-based probing. We form query–positive pairs $D_j = \{(q_i, d_i^+)\}_{i=1}^{|D_j|}$ using **CROP** or **Inverse Cloze Task** (ICT) (Lee et al., 2019; Izacard et al., 2021; González-Márquez et al., 2026). CROP pairs two distinct crops from one document, each containing two consecutive retained sentences. ICT pairs a sentence with its longer context, removing that sentence with probability 90% and retaining it otherwise. These probe short-text semantics and short-query/long-context matching, respectively. We use text-pair augmentations because dropout perturbations may vary with model depth (Gao et al., 2021).

Each module is evaluated by the negative InfoNCE loss (Oord et al., 2018):

$$s_j^{l,n} = \frac{1}{|D_j|} \sum_{i=1}^{|D_j|} \log \frac{\exp(d(\mathbf{z}^{l,n}(q_i), \mathbf{z}^{l,n}(d_i^+))/\tau)}{\exp(d(\mathbf{z}^{l,n}(q_i), \mathbf{z}^{l,n}(d_i^+))/\tau) + \sum_{b=1}^{M-1} \exp(d(\mathbf{z}^{l,n}(q_i), \mathbf{z}^{l,n}(d_{i,b}^-))/\tau)}. \tag{5}$$

Here $d(\cdot,\cdot)$ is cosine similarity and M is the number of candidates per query. The $M-1$ negatives $d_{i,b}^{-}$ are sampled from the positive-document pool outside the current query batch and shared within that batch. By default, $\tau = 0.05$ and module scores are averaged across three repeats. Scoring alternatives are compared in Figure 5b.

Downstream task-based probing. **TaskEval** uses labeled texts (x_i, y_i) for classification or clustering, scored pairs (x_i, x_i', y_i) for semantic similarity, or queries, candidate documents, and relevance judgments for retrieval or reranking. Let $\mathcal{E}_j(D_j; f)$ evaluate embedding function f on D_j, using accuracy, V-measure, Spearman correlation, NDCG@10, or MAP@1000 for the respective task types. Larger scores are better, and each module receives

$$s_j^{l,n} = \mathcal{E}_j(D_j; \mathbf{z}^{l,n}), \qquad (l,n) \in \mathcal{M}. \tag{6}$$

The LLM stays frozen and evaluators follow their task protocols. Both approaches yield $\pi_j = \operatorname{argsort}^{\downarrow}_{(l,n)\in\mathcal{M}} s_j^{l,n}$, where $\pi_j(a)$ is the module at rank a; ties follow layer and module indices. Modules share text forward passes (runtime comparison in Appendix A.3).

3.2 Static and Dynamic Routing

After scoring, routing assigns each task t an embedding function $x \mapsto \mathbf{z}_t(x)$, shared by all its examples. Static routes use assigned probing sources; dynamic routes select among dataset-specific experts using task instructions.

Static routing. Let $\mathcal{J}_t$ be the probing sources assigned to task t. CROP and ICT each assign their single source to all tasks; ICT/CROP assigns CROP to STS and Classification, and ICT to Clustering, Retrieval, and Reranking. Static TaskEval assigns the reference datasets from the target task's category and averages their module scores. Thus,

$$s_t^{l,n} = \frac{1}{|\mathcal{J}_t|} \sum_{j\in\mathcal{J}_t} s_j^{l,n}, \qquad \mathcal{S}_{t,k}^{*} = \operatorname{TopK}_{(l,n)\in\mathcal{M}}(s_t^{l,n}, k). \tag{7}$$

A single source selects the first k entries of π_j; TaskEval ranks the category-averaged scores. The text embedding is

$$\mathbf{z}_t(x) = \operatorname{Concat}(\{\mathbf{z}^{l,n}(x)\}_{(l,n)\in\mathcal{S}_{t,k}^{*}}) \in \mathbb{R}^{|\mathcal{S}_{t,k}^{*}| d_h}. \tag{8}$$

We concatenate modules in descending score order.

Dynamic routing. For each downstream reference dataset j, we retain its individual ranking π_j and define the expert $\mathcal{S}_{j,k}^{*} = \{\pi_j(1), \ldots, \pi_j(k)\}$. Given a target-task instruction I_t and reference instructions I_j, a frozen instruction encoder R selects the r most similar experts and computes their weights:

$$a_{t,j} = d(R(I_t), R(I_j)), \quad \mathcal{R}_r(t) = \operatorname{TopK}_j(a_{t,j}, r), \quad w_{t,j} = \frac{\exp(a_{t,j})}{\sum_{u\in\mathcal{R}_r(t)} \exp(a_{t,u})}. \tag{9}$$

For base LLMs, R is the eight-module CROP embedding. Routing runs once per task; $r = 1, 2$ give TaskEval (DR-1/DR-2). Appendix B.3 details instructions and tie-breaking.

For each task and module budget, let $\mathcal{C}_{t,k} = \bigcap_{j\in\mathcal{R}_r(t)} \mathcal{S}_{j,k}^{*}$ be the common modules. We concatenate their values as $\mathbf{z}_{\mathcal{C}_{t,k}}(x)$ in the top-ranked expert's order. For each selected expert j, let $\mathbf{z}_{j\setminus\mathcal{C}_{t,k}}(x)$ concatenate the remaining module values in their own π_j order. The task-specific text embedding is

$$\mathbf{z}_t^{\mathrm{DR}}(x) = \operatorname{Concat}\left(\mathbf{z}_{\mathcal{C}_{t,k}}(x), \sum_{j\in\mathcal{R}_r(t)} w_{t,j}\mathbf{z}_{j\setminus\mathcal{C}_{t,k}}(x)\right) \in \mathbb{R}^{kd_h}. \tag{10}$$

Common modules are preserved; residual vectors of dimension $(k - |\mathcal{C}_{t,k}|)d_h$ are mixed by rank position. DR-1 recovers one expert. Both routes need one forward pass per text.

4 EXPERIMENT

We first analyze the stability and task alignment of module rankings, then evaluate the embeddings they produce in training-free and post-processing settings.

4.1 EXPERIMENT SETUP

Unless otherwise specified, experiments use Llama3-8B (Grattafiori et al., 2024) and Qwen3-8B (Yang et al., 2025). CROP uses ICLR abstracts[1], following González-Márquez et al. (2026), while ICT uses Wikipedia[2].

TaskEval uses the following datasets and splits: **Clustering**, BiorxivClusteringP2P.v2 and MedrxivClusteringP2P.v2 (both test); **Reranking**, SciDocsRR (validation) and AskUbuntuDupQuestions (test); **STS**, STS15 (test) and STSBenchmark (validation); **Classification**, EmotionClassification and Banking77Classification (both train); and **Retrieval**, SciFact (train) and NFCorpus (validation, named `dev` in the source data). We average module scores across two datasets per category before ranking, using V-measure, MAP@1000, cosine Spearman, accuracy, and NDCG@10, respectively.

For CROP and ICT, we rank modules by their mean probing score across three repeats at $|D| = 20{,}000$ and $M = 256$.

4.2 ANALYSIS OF ATTENTION HEAD MODULE SORTING AND SELECTION RESULTS

Before comparing downstream embeddings, we examine whether probing budgets and data sources yield stable, task-relevant module rankings.

Robustness Analysis Figure 3 varies the probing-data size and candidate count within each fixed data distribution, comparing rankings with the same-repeat $(|D|, M) = (20{,}000, 256)$ reference. We measure full-ranking Spearman correlation and Top-8 overlap, the fraction of selected modules shared with the reference. Full rankings stabilize with modest budgets, while exact Top-8 membership varies more as the budget changes. For Qwen3-8B with CROP, these changes have limited impact on STS and Classification category averages, which remain above random selection (Appendix A.2).

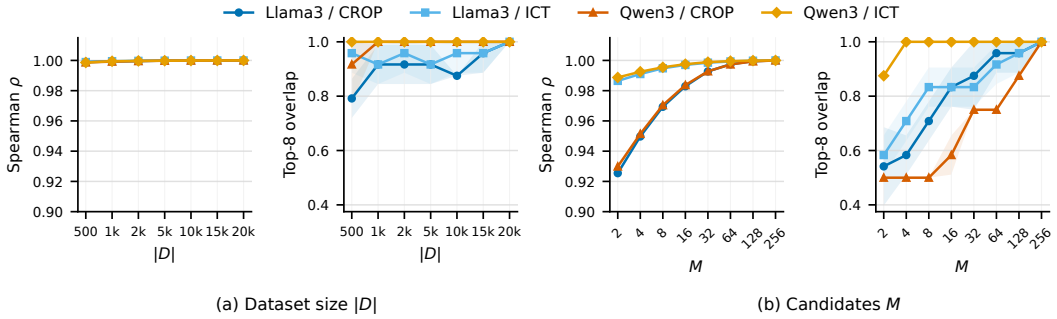


Figure 3: Attention head module ranking and selection stability. (a) $M = 256$; (b) $|D| = 20{,}000$. Curves compare with the same-repeat reference $(|D|, M) = (20{,}000, 256)$; shading shows ± 1 standard deviation over three repeats.

Cross-Method Similarity Analysis We next compare CROP and ICT with category-wise TaskEval rankings across five task categories (Zhang et al., 2026), using the datasets, splits, and score aggregation in Section 4.1. Figure 4 reveals complementary alignment: CROP is closer to STS, while ICT is closer to Clustering and Reranking. Retrieval favors different methods for full-ranking

[1] https://raw.githubusercontent.com/berenslab/iclr-dataset/37df25fa306c0cd97f023f11741f2d350e1de730/data/iclr24v1.parquet.gzip

[2] https://huggingface.co/datasets/wikimedia/wikipedia/tree/main/20231101.en

correlation and Top-8 overlap. Classification shows the weakest agreement, suggesting that generic augmentations miss label-specific distinctions. These patterns support task-specific selection, although differing source corpora prevent attributing them solely to augmentation. Appendix A.1 also shows module score maps and selections.

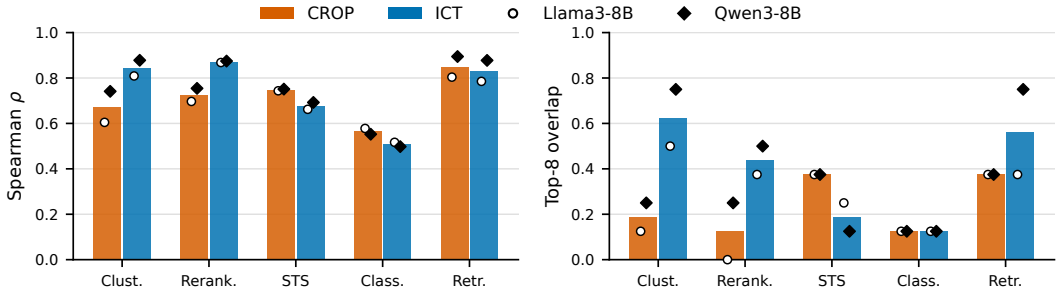


Figure 4: Agreement between augmentation-based and downstream attention head module rankings. Bars average Llama3-8B and Qwen3-8B; circles and diamonds show the respective model values.

4.3 Training-Free Setting

We now test whether the selected attention values yield competitive embeddings on MTEB tasks.

Baselines. We evaluate 14 MTEB tasks (Muennighoff et al., 2023), keeping final evaluation splits separate from TaskEval selection (Appendix B.1). Baselines include hidden-state (HS) and value (VA) averages over all layers (Full) or the upper half (Half), final-layer terminal-token pooling (LP), SGPT weighted pooling (WMP) (Muennighoff, 2022). We also compare the prompt methods in Section 2.1 and VA (LMP), which averages four downstream-selected layers (Zhang et al., 2026).

Selection and evaluation. All module-selection variants use eight modules per expert (1024 dimensions), with static or dynamic routing as in Section 3.2. TaskEval variants use the same ten reference datasets and raw values without final L2 normalization; Appendix B.3 details the dynamic router. Random averages three selections sampled without replacement from the model's upper half (seeds 13/42/73). Scores are scaled by 100; Avg. is the unweighted 14-task mean. Appendix B.1 specifies metrics and sampling details.

Results. Table 1 connects the module-ranking analysis to downstream performance. ICT/CROP exceeds the strongest baseline, MetaEOL, by 4.66/5.08 average points on Llama3/Qwen3 with one forward pass and unlabeled probing data. Its complementary experts improve over either augmentation alone and approach static TaskEval. The category scores explain this combination: ICT leads CROP on all three Clustering datasets and SciDocsRR for both models, while CROP leads on every STS and Classification dataset. These differences agree with the ranking patterns in Figure 4. DR-2 achieves the highest mean on both models, with task-level variation in its gains over static TaskEval and DR-1. This comparison evaluates dataset-specific expert selection and routing jointly. Complete Qwen3 results appear in Appendix B.2. CROP's retrieval advantage also depends on the source corpus; Appendix B.4 examines this alongside cross-method module overlap.

4.4 Ablation Studies

Figure 5 examines dimensionality, scoring, aggregation, and model scale using the same 14-task average as the main experiments. The first three experiments use Llama3-8B; the last uses Qwen3-0.6B, 4B, and 8B (Yang et al., 2025). Unless varied or using TaskEval, experiments use InfoNCE, fixed ICT/CROP routing, $|D| = 20{,}000$, $M = 256$, and three-repeat consensus.

Embedding dimension. We vary the selected attention head module count from 2 to 32, yielding 256–4096 dimensions (Figure 5a). Static TaskEval is the most stable across this range (57.29–57.81), whereas CROP and ICT are more sensitive to the module budget. ICT/CROP improves

Table 1: Training-free MTEB results. Each model lists baselines, then eight-module methods. Green marks the better ICT or CROP score; bold/underline mark the best/second-best mean. Random averages three seeds; its Avg. subscript is sample SD.

Model	Method	Dim.	Clustering			Retrieval			STS			Classification			Reranking		Avg.
			Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Llama3-8B	HS (Full)	4096	13.46	19.68	8.07	0.17	1.48	44.06	64.72	42.44	25.73	72.48	37.77	53.66	33.09	59.88	34.05
	HS (Half)	4096	19.62	22.59	10.04	6.24	1.48	44.76	65.95	44.61	30.92	72.07	37.24	58.17	33.93	61.22	36.35
	LP	4096	5.69	13.86	8.98	0.43	1.40	0.55	39.90	43.68	29.22	47.70	22.54	14.48	24.71	41.32	21.03
	WMP	4096	28.47	26.84	29.27	40.97	7.05	36.52	69.11	59.70	58.67	74.41	34.86	60.56	39.70	65.21	45.10
	EE	4096	29.47	27.14	35.11	35.11	8.70	23.03	71.72	65.95	65.59	79.23	39.28	62.35	42.88	73.29	47.06
	PromptEOL	4096	23.27	22.79	34.60	19.48	10.59	7.46	76.77	69.09	71.72	72.33	42.64	21.16	41.01	68.90	41.56
	PromptEOL+CP	4096	25.13	24.15	32.69	21.97	18.68	7.75	76.89	69.66	73.31	73.91	44.80	26.58	41.99	70.66	43.44
	MetaEOL	4096	33.37	29.71	45.18	32.62	14.63	23.51	80.78	75.72	78.18	81.38	50.70	60.49	41.30	78.92	51.89
	VA (Full)	1024	29.95	26.27	23.60	53.83	15.08	38.55	72.46	59.77	59.64	75.06	36.30	66.30	41.97	71.89	47.91
	VA (Half)	1024	30.73	26.25	23.28	58.09	18.67	40.96	71.99	59.63	59.99	74.81	35.36	68.35	41.99	71.28	48.67
	VA (LMP)	1024	33.75	29.74	30.69	60.03	23.02	40.84	75.82	63.14	64.29	76.28	38.40	63.67	43.87	76.29	51.42
	Random	1024	30.04	26.53	22.82	58.28	20.16	36.72	70.36	58.89	57.85	72.59	32.47	64.20	41.20	70.86	$47.35_{\pm 3.72}$
	ICT	1024	36.37	32.63	46.80	56.71	24.44	48.12	78.31	61.12	64.10	75.99	37.13	70.96	43.60	82.36	54.19
	CROP	1024	32.37	28.36	21.93	67.34	29.68	47.99	82.19	65.85	69.15	78.69	38.97	85.81	44.81	75.89	54.93
	ICT/CROP	1024	36.37	32.63	46.80	56.71	24.44	48.12	82.19	65.85	69.15	78.69	38.97	85.81	43.60	82.36	56.55
	TaskEval	1024	36.57	33.15	50.57	62.60	30.96	49.68	83.23	67.71	71.73	79.57	46.00	69.23	44.07	83.48	57.75
	TaskEval (DR-1)	1024	36.69	33.33	48.49	62.60	31.15	49.65	82.75	67.09	70.60	81.33	46.62	59.33	45.91	84.06	57.11
	TaskEval (DR-2)	1024	**36.62**	**33.75**	**49.12**	**62.54**	**30.65**	**49.48**	**83.38**	**67.11**	**71.46**	**81.09**	**46.59**	**71.65**	**46.91**	**84.44**	**58.20**
Qwen3-8B	MetaEOL	4096	29.63	26.95	36.33	43.84	13.12	21.81	80.82	74.66	72.79	83.28	54.11	72.71	41.35	77.66	52.07
	VA (LMP)	1024	33.68	29.26	28.00	60.76	22.93	41.84	75.66	62.27	63.89	75.78	37.32	75.65	44.89	76.91	52.06
	Random	1024	27.89	26.48	25.59	58.33	19.71	40.80	72.03	58.69	59.04	73.34	33.85	78.56	43.88	72.50	$49.34_{\pm 2.34}$
	ICT	1024	36.48	33.59	47.47	59.80	26.62	50.15	76.98	61.47	63.66	76.85	37.20	76.25	44.75	84.15	55.39
	CROP	1024	29.43	27.68	21.41	65.66	28.21	47.20	80.41	63.44	68.26	79.31	37.64	87.99	47.39	75.14	54.23
	ICT/CROP	1024	36.48	33.59	47.47	59.80	26.62	50.15	80.41	63.44	68.26	79.31	37.64	87.99	44.75	84.15	57.15
	TaskEval	1024	36.66	33.20	47.28	62.77	29.33	50.43	79.00	62.29	67.99	80.31	44.91	79.84	46.79	85.33	57.58
	TaskEval (DR-1)	1024	36.66	33.20	47.28	65.72	30.73	49.14	79.80	62.91	68.76	80.93	45.99	63.37	47.76	84.10	56.88
	TaskEval (DR-2)	1024	**36.66**	**33.20**	**47.28**	**65.03**	**30.66**	**50.19**	**80.23**	**63.04**	**68.84**	**80.95**	**44.50**	**75.47**	**48.48**	**84.04**	**57.75**

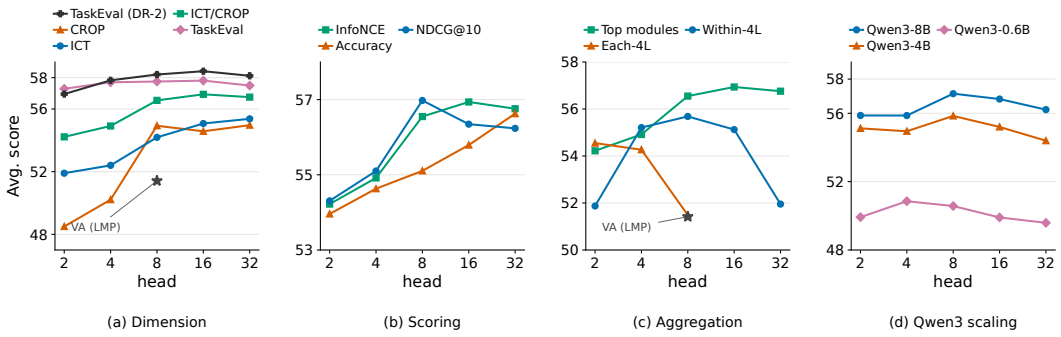


Figure 5: Training-free ablations: (a–c) Llama3-8B; (d) Qwen3 scaling. Scores average 14 tasks. Stars denote VA (LMP); ticks count selected attention head modules (per layer for Each-4L).

over either augmentation alone at every tested dimension, reaching 56.55 at eight attention head modules; increasing to 16 attention head modules adds only 0.39 points. TaskEval (DR-2) improves over static TaskEval from 512 to 4096 dimensions, reaching 58.20 at 1024 and peaking at 58.41 at 2048 dimensions. Eight modules therefore remain a compact operating point, although the dynamic curve peaks at a larger budget. This budget matches the eight-module methods in Table 1, enabling comparison at equal dimensions.

Scoring metric. We compare InfoNCE with positive-pair retrieval accuracy and NDCG@10 on the same augmented candidate sets (Figure 5b). InfoNCE consistently exceeds accuracy and achieves the best scores at 2048 and 4096 dimensions. NDCG@10 is slightly better at 256–1024 dimensions, including a 0.43-point advantage at 1024 dimensions. The pattern makes InfoNCE a competitive default across the full range of module budgets. Because the three metrics score the

same augmented candidate sets, their different curves show how the choice of ranking objective interacts with the number of modules retained for the final embedding.

Embedding aggregation. We compare global attention value module selection (Top modules) with restricting selection to four layers ranked by their complete-layer InfoNCE scores. Within-4L concatenates the value embeddings of the best k attention head modules across these four layers; Each-4L concatenates those of k attention head modules per layer and averages the four vectors, so its axis denotes modules per layer and its curve ends at $k = 8$ (Figure 5c). Global Top modules is best from 1024 dimensions onward: at 1024 dimensions it scores 56.55 versus 55.68 for Within-4L and 51.49 for Each-4L. This favors task-specific module selection across the model. The four-layer variants concentrate capacity within a smaller depth range, while global selection can use whichever individual modules rank highest for the assigned probing source.

Model scaling. With the same selection and routing procedure, Qwen3-8B and 4B peak at 1024 dimensions, scoring 57.15 and 55.84, respectively (Figure 5d). Qwen3-0.6B peaks earlier at 512 dimensions (50.85), with 1024 dimensions only 0.28 points lower. Across these model sizes, performance largely saturates by 512–1024 dimensions, with little gain from additional modules. Together, the ablations support the chosen scoring and module budget across model scales.

4.5 Post-Processing Setting

We evaluate task-specific dimensionality reduction on LLM2Vec-Llama3-Sup (BehnamGhader et al., 2024), using 2–32 modules with static TaskEval, ICT/CROP, or dynamic routing. All methods use task instructions at evaluation. The TaskEval comparison in Appendix C.1 shows higher mean scores with the instruction-aware extraction pipeline at every tested dimension. We therefore also apply the corresponding instructions when scoring modules on all ten TaskEval reference datasets, following LLM2Vec's input and pooling conventions. PCA uses the same reference datasets and splits, selecting components by variance.

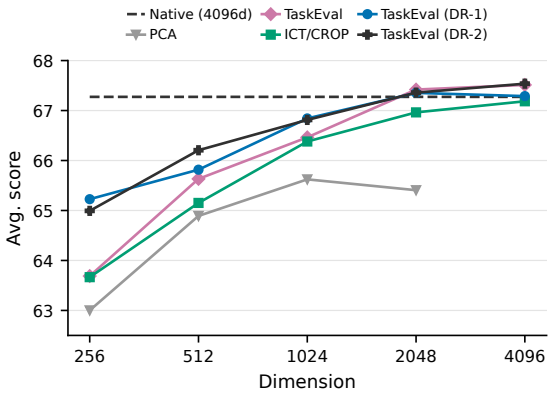


Figure 6: LLM2Vec dimension reduction with task instructions. Dashed line: native 4096d.

Figure 6 shows that all four module-selection variants outperform PCA at every reduced dimension (256–2048). At 256 dimensions, DR-1/DR-2 reach 65.23/64.99, compared with 63.69 for static TaskEval and 63.00 for PCA. At 1024 dimensions, they score 66.84/66.81, within 0.46 points of the native 4096-dimensional embedding (67.27). Dynamic routing is most useful at reduced dimensions, while static TaskEval remains competitive at higher dimensions. These results support task-specific module selection as an effective way to compress pretrained embeddings. Appendix C reports full scores and protocol details.

5 Conclusion

Probing task-specific attention head modules is a promising way to use existing attention values for text embedding. Module rankings can be obtained offline, and static selections or task-level routes are reused during encoding, so each text still requires just one model forward pass. This makes the method plug-and-play for frozen models: offline selections determine which existing modules produce compact, task-adapted representations. The approach also works on a trained LLM2Vec embedder, surpassing its native embedding at the same dimensionality and even with half as many dimensions. Together, these results support attention head module selection as an effective method for task specialization and dimensionality reduction. Future work will study how to incorporate attention values into embedding-model training.

Reproducibility Statement

Section 3 describes the augmentations, module scoring, and embedding construction. Sections 4.1–4.3 specify model and corpus sources, probing datasets and splits, and evaluation protocols. Appendix A provides selected module coordinates, robustness results, and scoring costs; Appendix B lists evaluation datasets and sample counts, complete training-free results, dynamic-routing details, and corpus comparisons. Appendix C details extraction, instructions, and dimension reduction.

Ethics Statement

This study uses existing pretrained models and public text datasets and does not involve recruiting human participants or collecting new personal data. Selected representations may inherit biases and encode sensitive information from their source models and corpora; improvements on embedding benchmarks do not establish fairness or privacy protection. Applications should respect the licenses and access conditions of the underlying models and datasets and assess deployment risks.

AI Use Statement

We used generative AI tools to assist with code implementation and the analysis and interpretation of experimental results. We also used these tools for manuscript drafting and language editing, and for preparing figures and tables. The authors reviewed the AI-assisted work and manually verified the experimental settings, code, reported results, and their interpretation. We take responsibility for the final content of this work, including AI-assisted text, claims, and artifacts.

References

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4895–4901, 2023.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 749–775, 2024.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Miguel Romero Calvo, Shuoyang Ding, Corey D Barrett, Georgiana Dinu, and George Karypis. Beyond instruction-conditioning, mote: Mixture of task experts for multi-task embedding models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22731–22746, 2025.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

Zifeng Cheng, Zhonghui Wang, Yuchen Fu, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. Contrastive prompting enhances sentence embeddings in llms through inference-time steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3475–3487, 2025.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, pp. 16124–16170, 2023.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6894–6910, 2021.

Rita González-Márquez, Philipp Berens, and Dmitry Kobak. Cropping outperforms dropout as an augmentation strategy for self-supervised training of text embeddings. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=gVRsIh9x7W.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Katharina Hämmerl, Alina Fastowski, Jindřich Libovickỳ, and Alexander Fraser. Exploring anisotropy and outliers in multilingual language models for cross-lingual semantic sentence similarity. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 7023–7037, 2023.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 3182–3196, 2024.

Sajjad Kachuee and Mohammad Sharifkhani. Geometry-preserving aggregation for mixture-of-experts embedding models. *arXiv preprint arXiv:2602.14039*, 2026.

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 *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 6086–6096, 2019.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10141–10157, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Mingxin Li, Zhijie Nie, Yanzhao Zhang, Dingkun Long, Richong Zhang, and Pengjun Xie. Improving general text embedding model: Tackling task conflict and data imbalance through model merging. *arXiv preprint arXiv:2410.15035*, 2024.

Xianming Li and Jing Li. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 792–804, 2024.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Ziyue Li and Tianyi Zhou. Your mixture-of-experts llm is secretly an embedding model for free. In *International Conference on Learning Representations*, volume 2025, pp. 70029–70046, 2025.

Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*, 2022.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

Zach Nussbaum and Brandon Duderstadt. Training sparse mixture of experts text embedding models. *arXiv preprint arXiv:2502.07972*, 2025.

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

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 459–477, 2024.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: uncovering hidden representations in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 55854–55875, 2025.

Jacob Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *International Conference on Learning Representations*, volume 2025, pp. 93543–93579, 2025.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1102–1121, 2023.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the twelfth language resources and evaluation conference*, pp. 4003–4012, 2020.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Bowen Zhang, Zixin Song, Chunquan Chen, Qian-Wen Zhang, Di Yin, and Xing Sun. Codiemb: A collaborative yet distinct framework for unified representation learning in information retrieval and semantic textual similarity. *arXiv preprint arXiv:2508.11442*, 2025a.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. Query-focused retrieval heads improve long-context reasoning and re-ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23802–23816, 2025b.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025c.

Yeqin Zhang, Yunfei Wang, Chen Jiaxuan, Ke Qin, Yizheng Zhao, and Cam-Tu Nguyen. LLM-based embeddings: Attention values encode sentence semantics better than hidden states. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=nAfV9H2Z1u`.

A Supplementary Analysis of Attention Head Module Sorting and Selection

A.1 Attention Head Module Score Maps

Figures 7 and 8 visualize the scores underlying the cross-method comparison. For visualization only, scores are independently min–max normalized within each panel after negating the contrastive loss for CROP and ICT, so brighter cells always indicate better-ranked modules. Rows and columns denote layers and distinct KV modules; layer 1 is at the bottom, and black borders identify each panel's Top-8 modules. The downstream panels use the category-wise mean scores and data splits in Section 4.1. The maps illustrate the same category-dependent pattern: CROP aligns more closely with STS, ICT with Clustering and Reranking, while Classification has weaker agreement and Retrieval differs between full-ranking and Top-8 comparisons. The separate color scales support comparison of spatial patterns across panels; absolute score magnitudes have panel-specific scales. Table 2 lists the eight marked modules for each scoring configuration in rank order. CROP and ICT use contrastive loss; the other five configurations use the corresponding downstream category scores.

The marked sets make this relation concrete. CROP shares three Top-8 modules with the STS category on each model. ICT shares four with Clustering on Llama3-8B and six on Qwen3-8B (Table 2). These overlaps complement the full-ranking correlations in Figure 4: related module orders can yield different compact representations when the budget is fixed at eight. Their model dependence also motivates examining both ranking correlation and selected-set overlap in the robustness analysis.

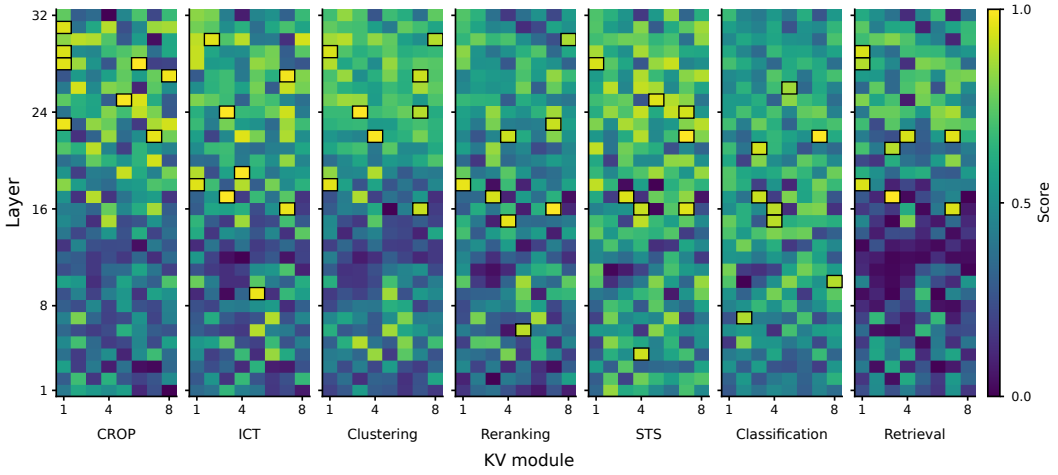


Figure 7: Llama3-8B attention head module scores. Colors show within-panel normalized scores; black borders mark the Top-8.

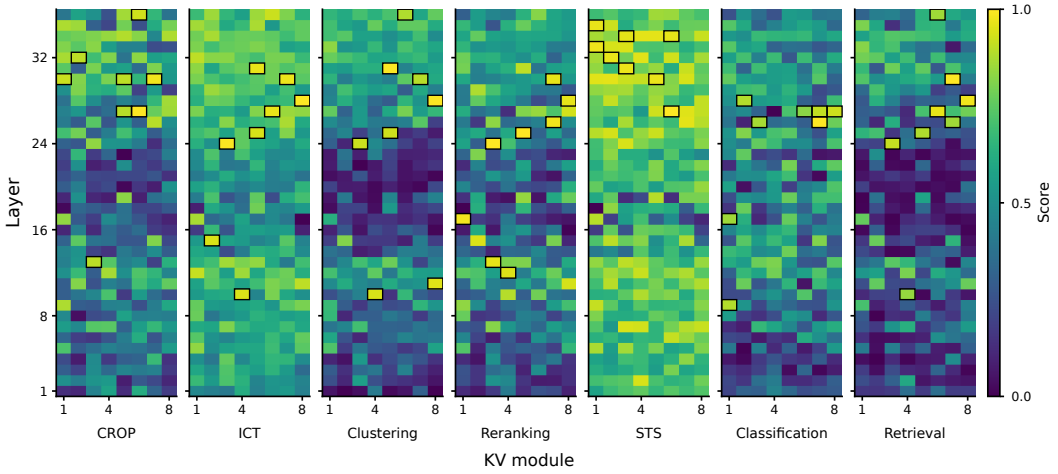


Figure 8: Qwen3-8B attention head module scores, using the same normalization and Top-8 marking as Figure 7.

Table 2: Top-8 attention head modules marked in the score maps, ordered from best to eighth-best. Each pair is (layer, KV module), with both indices starting at 1.

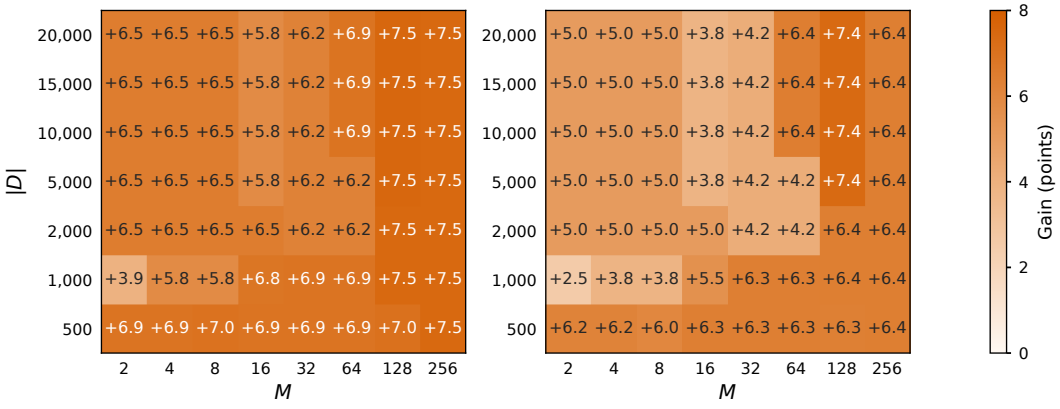
Scoring	1	2	3	4	5	6	7	8
Llama3-8B								
CROP	(28,6)	(25,5)	(27,8)	(23,1)	(22,7)	(28,1)	(31,1)	(29,1)
ICT	(17,3)	(24,3)	(19,4)	(27,7)	(9,5)	(16,7)	(30,2)	(18,1)
Clustering	(24,3)	(22,4)	(18,1)	(29,1)	(27,7)	(16,7)	(24,7)	(30,8)
Reranking	(16,7)	(18,1)	(15,4)	(17,3)	(22,4)	(23,7)	(6,5)	(30,8)
STS	(22,7)	(25,5)	(16,7)	(17,3)	(28,1)	(24,7)	(4,4)	(16,4)
Classification	(22,7)	(21,3)	(17,3)	(15,4)	(16,4)	(10,8)	(7,2)	(26,5)
Retrieval	(17,3)	(22,7)	(29,1)	(16,7)	(18,1)	(22,4)	(21,3)	(28,1)
Qwen3-8B								
CROP	(27,6)	(30,7)	(36,6)	(30,1)	(30,5)	(32,2)	(13,3)	(27,5)
ICT	(28,8)	(24,3)	(30,7)	(31,5)	(27,6)	(10,4)	(15,2)	(25,5)
Clustering	(31,5)	(28,8)	(11,8)	(10,4)	(25,5)	(24,3)	(30,7)	(36,6)
Reranking	(25,5)	(24,3)	(17,1)	(26,7)	(28,8)	(30,7)	(13,3)	(12,4)
STS	(27,6)	(33,1)	(30,5)	(31,3)	(32,2)	(35,1)	(34,6)	(34,3)
Classification	(26,7)	(27,8)	(27,7)	(28,2)	(9,1)	(26,3)	(17,1)	(27,6)
Retrieval	(30,7)	(28,8)	(27,6)	(24,3)	(25,5)	(36,6)	(26,7)	(10,4)

A.2 Downstream Robustness of Top-8 Selection

Ranking stability protocol. The dataset-size sweep uses $|D| = 500$–$20{,}000$ at $M = 256$, and the candidate-count sweep uses $M = 2$–256 at $|D| = 20{,}000$. Scoring and encoding batches remain fixed at two queries and four texts. Smaller datasets and negative sets are nested prefixes of the maximum-budget samples. Curves in Figure 3 show means and one sample standard deviation across three repeats; CROP samples may overlap, while ICT has disjoint parent samples. The reference endpoint equals one by construction.

Downstream evaluation. We examine Qwen3-8B with CROP, which shows the largest Top-8 overlap variation in the candidate-count sweep of Figure 3. We evaluate all 56 combinations of $|D| \in \{500, 1000, 2000, 5000, 10000, 15000, 20000\}$ and $M \in \{2, 4, 8, 16, 32, 64, 128, 256\}$, selecting eight attention head modules from the mean contrastive-loss ranking across three repeats. We report STS (STS17, SICK-R, STSBenchmark) and Classification (Banking77, Emotion, SprintDuplicateQuestions), the categories assigned to CROP in our routing rule. The random baseline uniformly samples eight attention head modules without replacement from layers 19–36 and averages three runs with seeds 13, 42, and 73. Each heatmap cell subtracts this random baseline from the selected attention head modules' mean score over the three datasets in that category, in score points.

Despite changes in Top-8 membership, the category means remain relatively stable: their standard deviations across the 56 settings are 0.65 points for STS and 1.14 for Classification. Every setting outperforms the category-matched random baseline, with gains of 3.92–7.52 and 2.54–7.40 points, respectively (Figure 9). Thus, variation in the selected attention head modules preserves substantial downstream benefits on the task categories targeted by CROP.



(a) STS (b) Classification

Figure 9: Qwen3-8B CROP Top-8 gains over random selection on (a) STS and (b) Classification. Cells show category-mean gains in score points; the random baseline averages three runs.

A.3 Scoring Cost

Table 3 compares the complete scoring costs, including all three seeds (13, 42, and 73) for CROP and ICT. VA (LMP) follows the layer-selection approach of Zhang et al. (2026). ICT requires about 1.1 GPU-hours per model across three seeds, comparable to the 1.2 GPU-hours for VA (LMP). CROP is substantially faster at about 0.5 GPU-hours per model because its input texts are shorter. Module-level TaskEval takes approximately twice as long as VA (LMP). For the first three methods, forward encoding accounts for most of the runtime; constructing the all-layer attention-value cache for the ten TaskEval datasets alone takes about one hour.

Scoring method	Llama3-8B	Qwen3-8B	Avg. GPU-hours/model
VA (LMP)	1:12	1:14	1.2
CROP (Modules)	0:29	0:33	0.5
ICT (Modules)	1:06	1:10	1.1
TaskEval (Modules)	2:21	2:31	2.4

Table 3: Approximate scoring costs. Per-model times use h:mm; the final column averages GPU-hours per model across the two models.

B Supplementary Training-Free Experiments

B.1 Evaluation Datasets

Table 4 lists the 14 evaluation tasks and the sizes of the test splits used in our experiments with MTEB (Muennighoff et al., 2023). Counts follow the cached dataset revisions used for the reported results; STS17 includes only the English–English subset. For clustering, counts sum text occurrences across ten clustering sets and therefore include repeated texts across sets. For retrieval and reranking, the count unit is evaluated queries. SprintDuplicateQuestions is a pair-classification task, grouped under Classification in our result tables. We report V-measure for clustering, NDCG@10 for retrieval, Spearman correlation for STS, accuracy for Banking77 and Emotion, average precision for SprintDuplicateQuestions, and MAP@1000 for reranking. Random samples eight modules uniformly without replacement from layers 17–32 of Llama3-8B or 19–36 of Qwen3-8B using seeds 13, 42, and 73. Each selection is evaluated independently; task scores are averaged across seeds, and the Avg. subscript reports the sample standard deviation of the three 14-task means.

The TaskEval configurations in Section 4.1 use different task variants or splits from final evaluation. For shared dataset names, TaskEval uses STSBenchmark validation, Banking77 and Emotion train, SciFact train, NFCorpus validation (`dev`), and SciDocsRR validation; final scores use their test splits. This separation is at the task/split level: the original datasets are not globally deduplicated, and retrieval corpora can be shared across splits.

Table 4: The 14 MTEB evaluation datasets. All counts refer to the test split; units distinguish text occurrences, sentence pairs, and queries.

Task category	Dataset	Samples	Unit
Clustering	BiorxivClusteringS2S	75,000	Texts (10 sets)
	MedrxivClusteringS2S	37,500	Texts (10 sets)
	TwentyNewsgroupsClustering	59,545	Texts (10 sets)
Retrieval	SciFact	300	Queries
	NFCorpus	323	Queries
	ArguAna	1,406	Queries
STS	STS17 (English–English)	250	Pairs
	SICK Relatedness (SICK-R)	9,927	Pairs
	STSBenchmark	1,379	Pairs
Classification	Banking77Classification	3,080	Texts
	EmotionClassification	2,000	Texts
	SprintDuplicateQuestions	101,000	Pairs
Reranking	StackOverflowDupQuestions	2,992	Queries
	SciDocsRR	3,978	Queries

B.2 Complete Results on Qwen3-8B

Table 5 reports all baselines and both routing settings under the same metrics as Table 1. ICT/CROP improves over MetaEOL by 5.08 points and approaches eight-module static TaskEval within 0.43

points. TaskEval (DR-2) achieves the highest mean, 57.75, on Qwen3-8B. The column abbreviations denote, in order: BiorxivClusteringS2S, MedrxivClusteringS2S, TwentyNewsgroupsClustering, SciFact, NFCorpus, ArguAna, STS17, SICK-R, STSBenchmark, Banking77Classification, EmotionClassification, SprintDuplicateQuestions, StackOverflowDupQuestions, and SciDocsRR.

Table 5: Complete Qwen3-8B training-free results (Dim.: embedding dimension). Green marks the better ICT or CROP score; bold/underlined rows mark the best/second-best mean. Random averages three seeds; its Avg. subscript reports sample SD.

Method	Dim.	Clustering			Retrieval			STS			Classification			Reranking		Avg.
		Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Qwen3-8B: Baselines																
HS (Full)	4096	6.67	17.27	5.39	0.14	1.70	39.46	61.43	38.87	22.85	71.63	37.96	45.32	28.60	53.65	30.78
HS (Half)	4096	10.76	19.60	5.46	0.43	1.80	41.72	62.54	40.53	26.67	71.48	37.64	52.69	29.34	55.38	32.57
LP	4096	4.21	11.95	12.12	0.10	1.19	0.29	36.20	32.80	26.22	30.91	24.10	1.86	23.33	38.86	17.44
WMP	4096	18.22	21.03	15.14	3.93	1.54	22.34	51.14	44.10	34.11	73.76	35.84	54.57	31.98	49.73	32.67
EE	4096	16.13	20.82	15.51	2.69	1.42	19.52	62.84	56.33	46.70	79.65	41.66	60.02	34.42	52.22	36.42
PromptEOL	4096	23.98	23.09	31.28	24.04	13.50	10.21	74.30	67.29	71.04	72.89	47.86	30.04	40.38	72.74	43.05
PromptEOL+CP	4096	26.23	24.12	35.72	28.88	17.89	15.53	73.17	69.19	73.52	73.78	47.93	31.82	42.14	75.07	45.36
MetaEOL	4096	29.63	26.95	36.33	43.84	13.12	21.81	80.82	74.66	72.79	83.28	54.11	72.71	41.35	77.66	52.07
VA (Full)	1024	27.33	24.68	20.96	59.45	19.30	38.68	71.80	60.37	60.76	75.74	33.14	81.91	44.69	70.45	49.23
VA (Half)	1024	26.82	24.68	21.61	59.42	19.38	38.82	72.13	60.66	61.31	75.80	32.97	82.18	44.89	70.45	49.37
VA (LMP)	1024	33.68	29.26	28.00	60.76	22.93	41.84	75.66	62.27	63.89	75.78	37.32	75.65	44.89	76.91	52.06
Qwen3-8B: Attention Value Module Selection (8 Modules)																
Random	1024	27.89	26.48	25.59	58.33	19.71	40.80	72.03	58.69	59.04	73.34	33.85	78.56	43.88	72.50	$49.34_{\pm 2.34}$
ICT	1024	36.48	33.59	47.47	59.80	26.62	50.15	76.98	61.47	63.66	76.85	37.20	76.25	44.75	84.15	55.39
CROP	1024	29.43	27.68	21.41	65.66	28.21	47.20	80.41	63.44	68.26	79.31	37.64	87.99	47.39	75.14	54.23
ICT/CROP	1024	36.48	33.59	47.47	59.80	26.62	50.15	80.41	63.44	68.26	79.31	37.64	87.99	44.75	84.15	57.15
TaskEval	1024	36.66	33.20	47.28	62.77	29.33	50.43	79.00	62.29	67.99	80.31	44.91	79.84	46.79	85.33	57.58
TaskEval (DR-1)	1024	36.66	33.20	47.28	65.72	30.73	49.14	79.80	62.91	68.76	80.93	45.99	63.37	47.76	84.10	56.88
TaskEval (DR-2)	1024	**36.66**	**33.20**	**47.28**	**65.03**	**30.66**	**50.19**	**80.23**	**63.04**	**68.84**	**80.95**	**44.50**	**75.47**	**48.48**	**84.04**	**57.75**

B.3 Dynamic Routing Details

The ten experts reuse the TaskEval datasets, splits, and task metrics in Section 4.1. Each dataset is ranked independently, without category averaging; score ties are broken by ascending layer and module index. For each model, the router encodes task instructions with its default ICLR-CROP top-eight modules (1024 dimensions), selected at $|D| = 20{,}000$, $M = 256$ using three-repeat consensus. Instructions follow the public LLM2Vec configuration (BehnamGhader et al., 2024).[3]

Routing uses cosine similarity and a softmax temperature of one over the selected experts, with all ten datasets eligible regardless of category. We normalize whitespace and remove trailing punctuation (periods, colons, and semicolons) before encoding; 24 task/reference instructions yield 15 distinct strings. The route is fixed across examples and module budgets. STS15 and STSBenchmark have identical instructions: a fixed PCG64 seed-13 draw for each of STS17, SICK-R, and STS-Benchmark selects STSBenchmark for DR-1; DR-2 uses both with equal weights. Other routing ties follow the fixed reference-dataset order listed in the main text.

For each budget, common modules are aligned first and residual blocks are fused in score order, as in Section 3.2. The reported rows use raw expert and final representations. Only the router consumes task instructions; target texts use plain inputs. At eight modules per expert, both DR-1 and DR-2 produce 1024-dimensional embeddings. These settings reuse downstream probing data and differ from static TaskEval in both expert construction and routing.

B.4 CROP Corpus and Cross-Method Module Rankings

We examine whether the CROP corpus helps explain its strong retrieval results, particularly on SciFact. Table 6 compares ICT with CROP on ICLR abstracts and Wikipedia, using eight attention

[3] https://github.com/McGill-NLP/llm2vec/blob/main/test_configs/mteb/task_to_instructions.json

head modules (1024 dimensions) throughout. ICT and ICLR-CROP reuse the main results. The available Wiki-CROP run uses $|D| = 10{,}000$ and $M = 64$, whereas the default selectors use $|D| = 20{,}000$ and $M = 256$; both average module losses over three repeats. This comparison includes both corpus and selection-budget differences.

The default CROP modules also rank relatively highly under ICT. Ranking all modules by ascending mean ICT loss, the eight CROP modules have ranks $31, 47, 28, 66, 21, 12, 38, 14$ out of 256 on Llama3-8B and $5, 3, 9, 43, 77, 28, 18, 31$ out of 288 on Qwen3-8B, listed in CROP selection order. Thus, all lie within the top 27% under ICT (worst ranks: $66/256 = 25.78\%$ and $77/288 = 26.74\%$). This shared preference may help explain CROP's occasional retrieval and reranking advantages; the selections and downstream behavior can still differ.

On SciFact, replacing ICLR with Wiki reduces CROP's margin over the default ICT from $+10.63$ to -0.17 points on Llama3-8B and from $+5.86$ to $+1.27$ on Qwen3-8B. Scientific abstracts may better match SciFact's domain and matching patterns, while the different selection budgets leave the corpus effect unresolved. Overall performance is model-dependent: Wiki lowers the Llama3-8B mean from 54.93 to 53.51, but raises the Qwen3-8B mean from 54.23 to 55.09.

Table 6: MTEB-14 results with eight attention head modules. ICT and CROP ICLR use the default selection budget; CROP Wiki uses the earlier 10k/64 budget. Abbreviations follow Table 5.

Method	Dim.	Clustering			Retrieval			STS			Classification			Reranking		Avg.
		Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Llama3-8B																
ICT	1024	36.37	32.63	46.80	56.71	24.44	48.12	78.31	61.12	64.10	75.99	37.13	70.96	43.60	82.36	54.19
CROP ICLR	1024	32.37	28.36	21.93	67.34	29.68	47.99	82.19	65.85	69.15	78.69	38.97	85.81	44.81	75.89	54.93
CROP Wiki	1024	35.98	32.73	46.16	56.54	24.55	46.54	76.80	60.75	62.95	74.73	37.09	70.36	42.33	81.69	53.51
Qwen3-8B																
ICT	1024	36.48	33.59	47.47	59.80	26.62	50.15	76.98	61.47	63.66	76.85	37.20	76.25	44.75	84.15	55.39
CROP ICLR	1024	29.43	27.68	21.41	65.66	28.21	47.20	80.41	63.44	68.26	79.31	37.64	87.99	47.39	75.14	54.23
CROP Wiki	1024	36.17	32.37	46.93	61.07	28.53	49.10	76.44	60.12	62.09	76.26	37.24	77.33	44.13	83.48	55.09

B.5 Training-Free Cross-Lingual Evaluation

We extend training-free evaluation to 16 languages with frozen Llama3-8B-Instruct, using the MIRACL Hard Negatives development split (Zhang et al., 2023).[4] The language codes in Table 7 denote Arabic, Bengali, English, Spanish, Persian, Finnish, French, Hindi, Indonesian, Japanese, Korean, Russian, Swahili, Telugu, Thai, and Chinese. Each query retrieves passages in its own language. Evaluation uses the pooled hard-negative corpus at revision `d7d94fa4b946cec4a27c84653aa0cf6b33f74a3c`, exact cosine retrieval, and NDCG@10; the average weights all languages equally. Inputs are plain text without task instructions, truncated to 512 tokens. Value embeddings use mean pooling over non-terminal tokens, followed by concatenation of selected modules and whole-vector L2 normalization; no model parameters are updated.

Language-specific scoring. For unsupervised **ICT+CL**, we construct inverse-cloze pairs from CC-100, the monolingual corpus reconstructed using CCNet (Wenzek et al., 2020).[5] For each language, reservoir sampling over at most the first 1 GiB of its compressed file supplies 30,000 eligible documents, partitioned into three disjoint 10,000-document repeats. One sentence serves as the query and is removed from the positive document with probability 0.9. Modules are ranked by mean InfoNCE loss (CL) across seeds 13/42/73. **TrainEval** instead probes modules with InfoNCE on labeled MIRACL training query–positive pairs.[6] Because the smallest training set has only 868 queries, each repeat samples 800 queries without replacement and one annotated positive per query;

[4] `https://huggingface.co/datasets/mteb/MIRACLRetrievalHardNegatives`
[5] `https://data.statmt.org/cc-100/`
[6] `https://huggingface.co/datasets/miracl/miracl`

repeats may overlap. This limited source size prevents matching the main experiments' scoring budget. Both ICT and TrainEval use 64 queries per scoring batch and $M = 64$ candidates per query: one positive and 63 shared negatives drawn outside the current batch, with temperature 0.05. The three per-module losses are averaged before selection.

TaskEval selects modules by their retrieval NDCG@10 on the BelebeleRetrieval test split (Bandarkar et al., 2024).[7] At revision `979a211276faa22f671e69d096634193567cfd05`, each aligned language supplies 900 queries and 488 passages deduplicated by their source link. Each module is scored separately against this corpus, and the best $k \in \{2, 4, 8, 16, 32\}$ modules are selected per language. These are selection data; final results are measured on MIRACL dev. The four-layer VA baseline follows the layer-selection approach of Zhang et al. (2026), reusing the Retrieval TaskEval scores from the main experiments and averaging four complete layers; its selection is shared across languages.

Results. At 1024 dimensions, ICT+CL, TaskEval, and TrainEval score 16.04, 20.14, and 20.43, respectively, compared with 15.01 for the four-layer baseline. Their best averages are 18.94, 20.63, and 20.74 at 32 modules. TaskEval and TrainEval exceed the four-layer baseline at every tested module count; ICT+CL does so at four of five counts, with a variable dimensionality curve. TrainEval serves as an in-distribution reference using labeled IID training data that may be unavailable in practice. TaskEval on another dataset in the same language (Belebele) and unsupervised CCNet-ICT also achieve strong results. At 32 modules, they outperform the strongest baseline by 5.61 and 3.92 NDCG@10 points, respectively, supporting language-specific module selection with readily available alternative data.

Table 7: Training-free MIRACL Hard Negatives results (NDCG@10 $\times 100$). k is the number of selected modules. The four-layer baseline uses shared Retrieval TaskEval; the three module-selection groups are language-specific.

Method	Dim.	k	AR	BN	EN	ES	FA	FI	FR	HI	ID	JA	KO	RU	SW	TE	TH	ZH	Avg.
HS (Full)	4096	–	0.00	0.00	0.00	0.00	0.10	0.27	0.00	0.32	0.01	0.00	0.65	0.20	0.45	0.36	0.17	0.04	0.16
HS (Half)	4096	–	0.06	0.09	0.00	0.00	0.18	0.27	0.00	0.60	0.03	0.04	1.22	0.20	0.65	0.39	0.49	0.06	0.27
LP	4096	–	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.07	0.00	0.00	0.00	0.04	0.04	0.01
WMP	4096	–	1.64	1.74	3.94	5.37	2.72	4.40	2.27	4.21	1.74	1.63	7.56	2.54	2.38	1.36	6.17	2.22	3.24
VA (Full)	1024	–	6.27	11.74	6.26	6.82	7.80	9.83	3.80	15.43	4.07	7.41	18.27	7.48	6.59	9.94	14.71	5.95	8.90
VA (Half)	1024	–	9.18	16.79	9.15	9.55	10.92	13.70	6.25	19.86	6.92	9.73	20.33	10.29	8.11	12.58	19.32	8.93	11.98
VA (LMP)	1024	–	12.29	19.74	12.16	14.04	14.12	18.03	8.11	22.18	9.18	12.48	23.57	13.04	11.13	15.39	22.68	12.07	15.01
VA (ICT+CL)	256	2	14.44	20.91	21.68	16.11	16.50	21.96	12.81	26.86	5.60	18.07	25.05	13.81	16.20	7.75	18.82	12.72	16.83
VA (ICT+CL)	512	4	11.17	21.23	10.34	11.74	13.51	18.42	7.69	21.36	8.10	15.27	27.99	11.06	14.27	10.83	20.57	15.46	14.94
VA (ICT+CL)	1024	8	11.22	17.11	12.48	14.46	16.33	19.32	8.61	22.74	10.92	16.25	30.70	11.94	11.24	10.93	23.95	18.49	16.04
VA (ICT+CL)	2048	16	14.91	18.68	16.24	13.50	17.95	18.99	10.64	27.29	12.90	17.73	32.97	14.61	15.87	12.62	23.86	20.82	18.10
VA (ICT+CL)	4096	32	14.49	21.40	15.92	16.45	19.37	22.76	12.30	26.37	13.55	17.59	33.89	14.13	15.81	13.72	25.50	19.74	18.94
VA (TaskEval)	256	2	12.46	20.61	19.41	16.11	14.06	22.91	12.81	22.72	13.01	19.13	29.64	15.05	14.39	12.00	22.99	18.34	17.85
VA (TaskEval)	512	4	14.22	21.23	22.38	15.02	16.01	21.99	12.80	24.87	12.06	19.06	33.38	14.08	17.41	12.61	22.35	20.29	18.73
VA (TaskEval)	1024	8	16.95	22.33	20.17	17.24	17.37	23.32	13.40	28.40	13.01	21.62	32.29	17.39	18.84	13.86	24.03	22.07	20.14
VA (TaskEval)	2048	16	17.60	20.89	20.07	17.25	17.96	22.60	12.68	29.13	13.41	19.30	33.78	18.53	20.55	12.95	24.63	20.95	20.14
VA (TaskEval)	4096	32	18.17	21.14	21.31	19.74	21.01	22.60	12.63	27.71	13.73	18.25	33.96	17.65	20.16	15.85	26.33	19.80	20.63
VA (TrainEval)	256	2	15.10	20.61	25.83	16.11	14.38	22.92	12.81	22.72	13.01	19.13	33.30	13.73	15.95	12.00	22.47	18.34	18.65
VA (TrainEval)	512	4	16.43	21.23	21.25	15.86	16.01	21.99	12.05	24.87	13.56	20.39	35.58	16.34	16.10	11.07	26.04	20.30	19.32
VA (TrainEval)	1024	8	18.00	22.66	21.68	15.68	17.06	22.31	14.41	26.49	14.49	20.01	33.65	18.22	20.24	15.67	23.43	22.86	20.43
VA (TrainEval)	2048	16	18.74	20.06	19.85	17.34	15.54	26.88	13.60	27.39	14.84	18.88	34.53	18.22	18.69	13.95	27.82	21.64	20.50
VA (TrainEval)	4096	32	19.29	21.23	20.37	19.56	19.52	25.71	13.89	27.37	14.84	17.51	33.60	15.74	17.61	14.74	28.89	22.04	20.74

C Supplementary Post-Processing Experiments

C.1 LLM2Vec Extraction and Dimension Reduction

We use the supervised LLM2Vec-Llama3 checkpoint (BehnamGhader et al., 2024), combining the MNTP and supervised adapters with bidirectional attention. Native embeddings mean-pool the final hidden states. For module selection, we extract each attention head module's value vectors, pool

[7] `https://huggingface.co/datasets/mteb/belebele`

over tokens, and concatenate the selected modules in score order. Each module contributes 128 dimensions, and the checkpoint remains frozen.

Instruction-aware scoring and final evaluation use the task instructions released with LLM2Vec's MTEB configuration[8]. Inputs follow its Llama3 user-header format. Pooling follows the native skip-instruction rule: prefix and instruction tokens are excluded, while the post-marker text suffix, including the terminal token when retained, is pooled. The maximum input length is 512 tokens, with a 400-token document limit. Symmetric inputs receive instructions on both sides; retrieval and reranking use instructions on queries only. Instructions are enabled both when ranking modules and when evaluating the selected representations.

For augmented-data scoring, CROP uses the STSBenchmark instruction on both crops of ICLR abstracts; ICT uses the SciFact instruction on the Wikipedia query, leaving its document and negatives unprompted. These post-processing runs use one fixed seed-13 set of 10,000 pairs per method, 64 contrastive candidates, and temperature 0.05. Final evaluation switches to each target task's instruction. ICT/CROP uses CROP for STS and Classification, and ICT for Clustering, Retrieval, and Reranking.

Dynamic routing. We reuse the ten instruction-aware TaskEval reference datasets and their original splits, selecting modules independently for each dataset. For LLM2Vec, the instruction encoder R is its native 4096-dimensional mean-pooled representation, encoding each instruction as plain text. We use the Top-1/Top-2 instruction routes, temperature-one softmax, and common-module alignment described in Section 3.2. Target examples retain their task instructions and native skip-instruction pooling. The reported curves use raw module representations at both expert and output levels, matching static TaskEval. At a budget of k modules per expert, the fused output remains $128k$-dimensional.

Category-specific PCA. PCA uses the same ten probing datasets and splits in Section 4.1. After exact text deduplication, we sample up to 5,000 encoder inputs per dataset without replacement (seed 13); retrieval/reranking sampling balances queries and documents where possible. The resulting category counts are 10,000 for Clustering, Reranking, and Classification, 7,910 for STS, and 8,916 for Retrieval. We extract instruction-aware native embeddings before L2 normalization, center them, and fit one non-whitened randomized PCA with 2048 components per category. Lower-dimensional embeddings use the first 1024, 512, or 256 components of that same fit. PCA fitting uses probing splits exclusively. PCA and TaskEval thus share source datasets, while their selection criteria remain variance and downstream score, respectively.

Instruction comparison. Table 8 reports every curve in Figure 6, the individual CROP/ICT results, and TaskEval without instructions. Instruction-aware TaskEval improves the mean by 0.04, 1.11, 0.52, 0.95, and 0.97 points at 256, 512, 1024, 2048, and 4096 dimensions, respectively. The no-instruction run uses non-terminal mean pooling, whereas the instruction-aware run follows native skip-instruction pooling. These differences reflect the full extraction-and-selection pipelines, including pooling and instructions. Column abbreviations and task metrics follow Table 1.

The task scores clarify the mean differences between static and dynamic selection. At 1024 dimensions, DR-1 raises the 14-task mean from 66.46 to 66.84. Changes vary across tasks: SciFact moves from 75.13 to 74.89, while STS17 rises from 89.02 to 90.03 and ArguAna from 53.51 to 54.87. At 256 dimensions, DR-1 reaches 65.23, exceeding static TaskEval by 1.54 points; at 4096 dimensions, DR-2 and static TaskEval are close (67.54 and 67.51). The largest routing gain in the table thus occurs at the smallest embedding budget.

[8] https://github.com/McGill-NLP/llm2vec/blob/main/test_configs/mteb/task_to_instructions.json

Table 8: LLM2Vec-Llama3-Sup: complete MTEB-14 scores, including dynamic routing and individual augmentations. All rows use instructions except TaskEval (no inst.).

Method	Dim.	Clustering			Retrieval			STS			Classification			Reranking		Avg.
		Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Native	4096	36.96	32.89	55.37	77.50	41.42	60.64	89.57	83.94	88.04	87.92	51.57	95.10	54.88	86.04	67.27
PCA	256	37.23	34.36	46.52	75.48	39.84	49.41	89.76	82.43	86.58	84.43	44.14	79.93	47.09	84.85	63.00
PCA	512	36.96	34.10	52.13	76.17	41.19	55.48	91.23	83.55	88.30	83.16	41.71	88.36	50.67	85.48	64.89
PCA	1024	37.22	32.96	54.48	77.57	42.00	58.74	91.85	83.90	88.94	80.89	37.46	93.72	53.31	85.68	65.62
PCA	2048	37.04	33.09	55.21	77.61	41.96	59.42	92.01	83.92	89.15	76.61	33.76	95.92	54.19	85.79	65.41
CROP	256	35.12	35.35	49.59	69.63	34.52	39.15	81.75	80.84	84.95	87.02	38.01	90.97	53.18	86.02	61.86
CROP	512	37.09	35.57	55.30	73.32	37.50	51.70	87.16	83.43	86.62	87.77	43.17	91.04	54.74	86.92	65.10
CROP	1024	37.26	34.37	55.38	75.18	39.54	59.45	88.86	83.92	88.49	88.00	47.05	94.72	54.52	86.84	66.68
CROP	2048	36.29	33.83	54.62	76.33	40.32	60.79	89.65	84.39	88.98	88.10	48.54	94.70	55.30	86.40	67.02
CROP	4096	35.62	31.19	48.76	75.77	39.77	61.08	89.98	83.91	88.87	87.92	49.51	95.48	54.82	83.39	66.15
ICT	256	38.01	37.04	56.33	71.10	35.62	51.76	86.35	82.23	83.55	86.09	39.00	84.37	51.92	86.01	63.53
ICT	512	38.14	35.29	57.82	71.99	34.25	56.03	87.88	83.57	84.80	86.86	43.68	87.57	53.06	86.35	64.81
ICT	1024	38.56	35.71	57.00	73.00	35.27	57.95	88.61	83.76	84.73	87.30	44.39	86.69	54.07	86.73	65.27
ICT	2048	38.57	35.01	56.08	75.93	40.12	56.05	89.55	84.87	88.31	87.65	46.59	92.83	54.30	87.07	66.64
ICT	4096	37.16	33.85	55.50	76.48	41.40	59.14	90.78	84.94	88.46	87.32	48.56	91.85	55.10	86.30	66.92
ICT/CROP	256	38.01	37.04	56.33	71.10	35.62	51.76	81.75	80.84	84.95	87.02	38.01	90.97	51.92	86.01	63.67
ICT/CROP	512	38.14	35.29	57.82	71.99	34.25	56.03	87.16	83.43	86.62	87.77	43.17	91.04	53.06	86.35	65.15
ICT/CROP	1024	38.56	35.71	57.00	73.00	35.27	57.95	88.86	83.92	88.49	88.00	47.05	94.72	54.07	86.73	66.38
ICT/CROP	2048	38.57	35.01	56.08	75.93	40.12	56.05	89.65	84.39	88.98	88.10	48.54	94.70	54.30	87.07	66.96
ICT/CROP	4096	37.16	33.85	55.50	76.48	41.40	59.14	89.98	83.91	88.87	87.92	49.51	95.48	55.10	86.30	67.19
TaskEval	256	38.50	36.19	55.85	68.14	34.40	37.92	88.23	83.40	87.78	85.37	49.16	87.26	52.76	86.68	63.69
TaskEval	512	38.43	36.53	56.10	73.35	37.27	48.67	88.53	83.83	88.45	86.63	50.28	90.34	53.48	86.96	65.63
TaskEval	1024	38.34	35.72	57.08	75.13	39.53	53.51	89.02	83.65	88.85	87.48	51.02	89.66	54.94	86.55	66.46
TaskEval	2048	38.38	34.52	56.60	75.94	41.99	59.23	89.40	84.22	89.06	87.60	51.79	92.66	55.62	86.86	67.42
TaskEval	4096	38.02	33.90	56.54	76.71	41.73	59.44	89.62	83.76	88.89	87.71	52.24	94.92	55.26	86.45	67.51
TaskEval (DR-1)	256	38.45	37.26	56.35	70.97	34.41	51.60	88.20	83.39	87.77	87.25	49.09	87.74	53.62	87.06	65.23
TaskEval (DR-1)	512	38.32	36.51	55.92	73.51	37.27	55.08	89.56	84.09	87.65	87.79	49.63	84.87	54.32	86.92	65.82
TaskEval (DR-1)	1024	38.80	35.16	57.05	74.89	39.94	54.87	90.03	84.02	88.74	87.89	51.10	91.17	55.12	86.99	66.84
TaskEval (DR-1)	2048	38.38	35.32	57.27	76.14	40.79	59.29	89.32	84.14	88.98	88.08	51.91	91.00	55.25	87.07	67.35
TaskEval (DR-1)	4096	38.36	33.97	55.29	76.78	41.61	59.16	89.72	84.06	88.88	87.99	51.33	92.68	55.60	86.61	67.29
TaskEval (DR-2)	256	38.31	36.52	56.08	70.20	35.71	46.50	88.20	83.39	87.77	86.98	49.37	89.80	54.00	87.07	64.99
TaskEval (DR-2)	512	38.32	36.51	55.92	74.32	38.30	53.98	89.16	83.98	88.37	87.74	49.48	88.95	54.73	87.12	66.21
TaskEval (DR-2)	1024	38.52	35.95	56.78	74.94	39.62	53.18	89.47	84.08	88.79	88.00	50.54	93.40	55.17	86.94	66.81
TaskEval (DR-2)	2048	38.45	34.82	56.63	76.32	40.88	58.69	88.94	84.03	88.98	88.17	51.38	93.58	55.20	86.95	67.36
TaskEval (DR-2)	4096	38.53	34.50	56.34	76.71	41.85	59.76	89.62	83.91	88.89	87.87	51.27	94.18	55.36	86.75	67.54
TaskEval (no inst.)	256	38.59	37.03	54.12	67.62	32.41	40.43	88.22	81.76	85.93	85.38	50.01	90.29	52.74	86.55	63.65
TaskEval (no inst.)	512	38.62	37.01	55.07	70.66	34.65	43.83	89.25	82.54	86.72	86.06	50.54	88.53	52.91	86.85	64.52
TaskEval (no inst.)	1024	38.53	36.02	56.46	72.96	36.86	52.43	89.13	82.70	86.79	87.06	51.52	92.42	53.60	86.81	65.95
TaskEval (no inst.)	2048	38.06	35.00	55.95	74.73	39.54	54.55	89.14	82.78	86.93	87.21	51.71	93.72	54.65	86.62	66.47
TaskEval (no inst.)	4096	38.26	34.21	54.98	75.65	39.92	55.44	88.98	82.86	86.96	87.33	51.62	94.75	54.68	85.95	66.54