

MetaEncoder: Exploring the Limit of Bi-Encoders for Multimodal System One Decision Making with Natural Language Interface

Jianpeng Cheng, Guangyu Sun, Aashu Singh, Benyu Zhang, Haixing Dai, Hossein Mansour, Jiangfan Zhang, Shlok Kumar Mishra, Wei Sun, Xuanming Cui, Yanli Liu, Qi Guo, Max Xiangjun Fan, Jun Xiao

¹Meta AI

System One models output constrained decisions and probability distributions rather than free-form text generation. While prevailing paradigms rely on structured schema objects to encode state, intent, and candidate choices, we revisit a fully natural language-based System One interface. In this framework, both the user request and each candidate option are expressed in natural language, supported by multimodal (image and video) auxiliary inputs. We introduce MetaEncoder, which fine-tunes a pre-trained Muse-Glimmer 30B decoder into an instruction-following decision-making encoder. To scale effectively across both small closed-set (< 256) and massive open-set (millions) candidate spaces, MetaEncoder employs a bi-encoder architecture trained via unidirectional contrastive learning for request-candidate alignment. We conduct extensive evaluations across 11 benchmark suites and 190 tasks spanning multimodal decision-making, understanding (closed-set) and retrieval (open-set), highlighting where MetaEncoder beats SOTA multimodal encoders, as well as its current limits on reasoning-intensive tasks.

Date: October 9, 2026

Hugging Face: <https://huggingface.co/facebook/meta-encoder>

GitHub: <https://github.com/facebook/meta-encoder-eval>



1 Introduction

There is a recent trend of repurposing general-purpose encoders as *System One* models for decision making (TypeSafe AI, 2026). An encoder-only model answers fast and in a constrained form. It returns a choice among candidates, together with a probability distribution over them, instead of generating free-form text. Many production decisions fit this form: routing a request to an agent, selecting a tool, classifying content against a policy, ranking items, picking the next action in an environment, or retrieving a memory. For these, a calibrated score over candidates from a single forward pass is more useful than a generated answer.

Encoder models that score candidates are not a new topic. First, sentence-BERT (Reimers and Gurevych, 2019) and dense retrievers (Karpukhin et al., 2020) popularized *bi-encoders*. These embed a query and a candidate independently and compare them by vector similarity, and they were later scaled into universal text embedders (Wang et al., 2022; Xiao et al., 2024; Su et al., 2023; Wang et al., 2024). In vision, two-tower models such as CLIP (Radford et al., 2021) align images with texts through contrastive learning. More recently, multimodal embedders built on vision-language backbones cast classification, question answering, and grounding as retrieval over candidates (Jiang et al., 2024, 2025; Zhang et al., 2025a). Second, *Cross-encoders* (Nogueira and Cho, 2019; Nogueira et al., 2020) instead encode the query and the candidate together, trading efficiency for accuracy, and remain the standard choice for reranking. Finally, *constrained decoders* can also score candidates directly. The options are listed in the prompt under identifiers (A, B, C, ...), and the model picks the identifier with the highest next-token probability (Hendrycks et al., 2021; Robinson et al., 2023; Gangi Reddy et al., 2024). This model compares all options jointly, but the candidate cardinality is bounded by the context window and results are sensitive to option order (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024). We review all above work in Section 2.

When such a model is repurposed as a System One model, a common interface is a *structured request object*.

It contains separate fields for the context or state, the questions or instructions, the options and scoring criteria, and each task has its own schema. Schemas are precise, but they are rigid. In this work we revisit a fully *natural-language* interface. The prompt is free-form text, optionally with images or video, that describes the state, the instruction, and the criteria for the task. The output stays constrained to a set of candidate options, each of which is also described in natural language. Our goal is a general-purpose encoder that follows instructions and makes decisions, similar to prompting a decoder-only model.

A key consideration in choosing the encoder architecture is balancing efficiency against scalability. Scalability here means both the number of candidates that can be scored and whether candidate representations can be cached. Cross-encoders and constrained decoders are powerful because the request attends to candidates directly. However, they cannot cache efficiently, so they do not scale to large candidate sets. Bi-encoders encode candidates independently of the request. This makes the candidate cardinality scale from a closed set of <256 options to millions of items. However, the price is that all interaction between request and candidate passes through a single vector. Whether that bottleneck can still support instruction generalization is an open question. In this work we revisit bi-encoders and explore how far they can go as general-purpose, instruction-following encoders. Empirically, we demonstrate that bi-encoders can also achieve high accuracy in closed-set decision-making by incorporating candidate options directly into the request’s prompt, supplying the model with information equivalent to that of a constrained decoder.

We train **MetaEncoder**, a general-purpose multimodal encoder for instruction-following decision making, understanding and retrieval. MetaEncoder fine-tunes a pre-trained **Muse-Glimmer** (Meta Superintelligence Lab, 2026) 30B decoder into an encoder. For training, MetaEncoder adopts unidirectional contrastive learning to align each request with its target candidate, differentiating closed- and open-set problems through the negative selection strategy (i.e., using a provided option list vs. gathering across all devices). For inference, the same model handles small closed sets of fewer than 256 options and open corpora of millions of candidates, which differ only in how the request prompt is constructed. MetaEncoder adopts the natural-language interface described above, so users interact with it similar to a decoder-only LLM. Both the request and the candidates can be multimodal.

We evaluate MetaEncoder across 11 benchmark suites comprising 190 tasks. These span closed-set multimodal decision-making and understanding (JevBench (Standhartinger and Contributors, 2026), ImaJevBench (Mohit and Contributors, 2026), MMLU (Hendrycks et al., 2021), MMMU (Yue et al., 2024), VideoMMMU (Hu et al., 2025), NaturalBench (Li et al., 2024), and TempCompass (Liu et al., 2024)) as well as open-corpus retrieval (NanoBEIR (AI, 2025), MMEB-v3 image, video, and visdoc (Huang et al., 2026)). MetaEncoder outperforms state-of-the-art (SOTA) multimodal embedding models (Zhou et al., 2026; Li et al., 2026) on all decision-making tasks while achieving near-SOTA results on multimodal retrieval benchmarks which is beyond the scope of Jev models (Wortega and Contributors, 2026). On the other hand, we also identify the inherent limitations of bi-encoder architectures on complex decision-making tasks relative to decoders, where the lack of explicit reasoning capabilities remains the primary bottleneck.

In summary, our contributions are:

- We revisit a fully natural-language System One interface. The request is free-form text with auxiliary images and video, and the output is constrained to candidate options that are themselves described in natural language.
- We train MetaEncoder, a bi-encoder fine-tuned from a 30B multimodal decoder with request-to-candidate contrastive learning. It scales from closed sets of fewer than 256 options to open corpora of millions of candidates.
- We conduct an extensive evaluation over 11 benchmark suites and 190 tasks. It identifies both where a bi-encoder can act as a general-purpose, instruction-following encoder and where it does not yet generalize.

2 Related Work

A general-purpose encoder maps an *instruction*, a *context* (text, image, video, etc.), and a set of *candidates* to a decision. Classification, retrieval routing, reranking, and judgment all fit this one interface. They differ only in where the candidate set comes from: a global corpus, a fixed label space, or a set defined by each example, as in Jev-style tasks where every row carries its own options and criteria. Early universal embedders focused on text retrieval and semantic similarity (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Wang et al., 2022; Li et al., 2023; Xiao et al., 2024). Later work added instructions to the input (Su et al., 2023; Wang et al., 2024; Lee et al., 2025; Muennighoff et al., 2025; BehnamGhader et al., 2024) and extended the recipe to multimodal inputs using vision-language backbones (Jiang et al., 2024, 2025; Zhang et al., 2025a; Lin et al., 2025; Gu et al., 2026; Chen et al., 2026; Zhang et al., 2025a), and augmenting encoder input with reasoning (Cui et al., 2026, 2025; Lan et al., 2026; Cheng et al., 2026; Jiang et al., 2026). Benchmarks such as MTEB (Muennighoff et al., 2023) and MMEB (Jiang et al., 2025) already cast classification, VQA, and grounding as ranking over candidates. Broadly, a model built on a language-model backbone can score candidates in 3 ways. They differ in how much the prompt and the candidates interact, and therefore in cost, scalability, and expressiveness:

(1) Bi-encoders. A bi-encoder embeds the prompt (instruction, state, context, etc.) and each candidate separately and scores them by cosine similarity. It is trained with a CLIP-style contrastive objective that pulls the prompt toward its gold candidate and pushes it away from in-batch or hard negatives (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023; van den Oord et al., 2018). This is the dominant design for universal embedders (Wang et al., 2022; Su et al., 2023; Wang et al., 2024; Lee et al., 2025; Jiang et al., 2025; Zhang et al., 2025a). Its main advantage is efficiency: since candidate embeddings do not depend on the prompt, they can be precomputed and indexed. Scoring K candidates then costs a single prompt forward pass plus K dot products, and this scales to candidates of millions. However, the tradeoff lies in expressiveness: since all interaction happens through a single dot product, the prompt embedding must anticipate every candidate it might be compared against. This is a known weakness on fine-grained or compositional distinctions (Yuksekgonul et al., 2023; Thrush et al., 2022). It is especially limiting when what a label means is defined by the prompt itself, for example by row-specific criteria, since the candidate embedding cannot see those criteria. Late-interaction models such as ColBERT (Khattab and Zaharia, 2020) relax the single-vector bottleneck but still encode the two sides independently.

(2) Cross-encoders. A cross-encoder concatenates the prompt with *one* candidate at a time. A shared transformer processes the pair jointly, and a scalar relevance score is read from a classification head or from the logit of a *yes/true* token (Nogueira and Cho, 2019; Nogueira et al., 2020; Ma et al., 2024; Zhang et al., 2025b). Full token-level attention between prompt and candidate makes cross-encoders consistently more accurate than bi-encoders. That is why they serve as the second-stage reranker in retrieve-then-rerank pipelines (Nogueira and Cho, 2019; Xiao et al., 2024; Cui et al., 2025). LLM-as-a-judge pointwise scoring follows the same pattern (Zheng et al., 2023). Two costs follow from the design: First, nothing can be cached: scoring K candidates takes K full forward passes over the shared prompt. Second, each candidate is scored in isolation, so the model never compares candidates directly. Scores must be calibrated across independent passes, and the model cannot use the structure of the option set (none of the above, mutually exclusive labels, near-duplicate options) to decide.

(3) One-step constrained decoding. The third approach puts the instruction, the context, and *all* candidates into a single prompt. Each candidate is labeled with an identifier (A, B, ..., Z), and a generative model is asked to emit one identifier token. The prediction is the argmax of the next-token distribution restricted to the valid identifiers. Since the vocabulary projection can be pruned to constrain outputs, the approach acts as a cheap encoder prefilling pass topped with a light linear decision layer. This is the standard multiple-choice format for evaluating LLMs (Hendrycks et al., 2021; Robinson et al., 2023). It underlies zero-shot classification through verbalizers (Schick and Schütze, 2021) and has been adopted for listwise reranking, where the first generated identifier’s logits act as the ranking score (Sun et al., 2023; Gangi Reddy et al., 2024). Because all candidates share one context, the model can compare them jointly. Inference takes a single forward pass regardless of K , and no new head or training objective is needed beyond the language-modeling loss. On the other hand, the limitations come from treating candidates as tokens in a sequence. Accuracy is sensitive to option order and identifier choice (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024). The candidate set is

bounded by the size of the identifier alphabet and by the context window, so the approach cannot support large candidates. Besides, the output is a distribution over positions, but not a reusable representation of the candidates. As a result, nothing can be indexed, cached across prompts, or reused for retrieval and clustering.

Summary. The three designs trade off along one axis: how much the prompt and the candidates interact. Bi-encoders allow none interactions, which buys indexing and scale. Cross-encoders let the prompt interact with each candidate individually, which buys accuracy at $O(K)$ cost and no representation caching. Constrained generation lets all candidates interact with each other and with the prompt in one pass, but it gives up open-ended candidate sets and reusable embeddings completely. In this work, we adopt a cachable bi-encoder design to explore its limit in general-purpose decision making.

3 MetaEncoder

3.1 Natural-language interface

MetaEncoder scores a set of candidates $\mathcal{C} = \{c_1, \dots, c_K\}$ against a request q . The request is a free-form prompt, optionally interleaved with images or video. It describes everything the decision depends on, such as the context and state; the instruction and question; the criteria which define what each option means. Each candidate c_k is also expressed in natural language: a text description with optionally an image or video. Closed-set decision making and open-set retrieval therefore share one interface. They differ only in whether $\mathcal{C}$ is a small set which can be added into the request as additional context, or a large corpus which has to be indexed in advance. For large open-set candidates, it is not possible to add them as additional context into the request due to context window restrictions.

3.2 Architecture

MetaEncoder is a bi-encoder initialized from the pre-trained **Muse-Glimmer** 30B multimodal decoder. The same network f_θ encodes requests and candidates. Images and video are processed by the backbone’s native vision tower and interleaved with the text tokens.

Last-token representation. We keep the backbone’s causal attention and take the final-layer hidden state of the *last* token as the representation, normalized to unit length:

$$\mathbf{q} = \frac{h_{\text{last}}(q)}{\|h_{\text{last}}(q)\|_2}, \quad \mathbf{c}_k = \frac{h_{\text{last}}(c_k)}{\|h_{\text{last}}(c_k)\|_2}. \quad (1)$$

Under causal attention, the last token attends to every token in the input. It therefore sees the full context for the request, including state, instruction, criteria and options (for closed-set only), as well as all visual tokens, and can therefore summarize the request.

Candidates in the prompt. For small closed sets, the candidate options can be listed in the request itself. Listing them in request gives the last token a global view, so that the model sees which alternatives exist and can contrast options. This recovers some of the joint-reasoning-over-candidate ability offered by cross-encoders and decoder-only models. However, for open-set retrieval, the candidate pool is too large to list in request anyway. To support both cases, during training we dropout the option list from the prompt for a random fraction of closed-set examples. The model then cannot purely rely on matching option text in the prompt and must still embed the request meaningfully.

Scoring. Each request-candidate is scored by cosine similarity, $s(q, c) = \mathbf{q}^\top \mathbf{c}$. So a softmax over $\{s(q, c_k)/\tau\}_{k=1}^K$ gives a probability distribution over the candidate set. In this way, candidate embeddings do not depend on the request, so they can be computed once and cached. This scoring mechanism scales from a closed set of a few options to an index of millions of items.

3.3 Unidirectional Contrastive Training

We train MetaEncoder to align each request with its correct candidate. Each example is a pair (q_i, c_i^+) . For a batch of B requests, the loss is an InfoNCE objective (van den Oord et al., 2018) applied in the

request-to-candidate direction only:

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(s(q_i, c_i^+)/\tau)}{\sum_{c \in \mathcal{N}(q_i) \cup \{c_i^+\}} \exp(s(q_i, c)/\tau)}, \quad (2)$$

The negative set $\mathcal{N}(q_i)$ depends on the training task type:

- **Open-set (retrieval).** $\mathcal{N}(q_i)$ contains the gathered candidates of the other requests in the global batch, plus hard negatives where available. Any in-batch candidate that is identical to c_i^+ is removed from $\mathcal{N}(q_i)$.
- **Closed-set (decision making).** $\mathcal{N}(q_i)$ is the request’s own option set, minus the gold option. The softmax therefore runs over exactly the candidates the task offers, so training matches evaluation, where each example is scored against its own options.

4 Experiment

4.1 Benchmarks

We evaluate MetaEncoder on 11 benchmarks comprising 190 tasks (Table 1). We group them into three capabilities: decision making, multimodal understanding, and retrieval. They span both candidate regimes: in the *closed-set* regime, each request comes with its own small option set (fewer than 256 candidates). In the *open-set* regime, the request is scored against a shared larger candidate pool.

Decision making. These benchmarks are recently established to evaluate System One models. Each example in the benchmarks consists of a task state, an instruction, decision criteria that define the labels, and a set of label options.

- **JEVBench** (Standhartinger and Contributors, 2026) contains 231 text-only decision rows in three splits: *original* (72 rows), *easy* (48), and *hard* (111). Each row defines its own label set of 2 to 6 options, and the meaning of each label is given only by the row’s criteria. The splits contain 6, 13, and 80 distinct task instructions, respectively.
- **ImaJEV-Bench** (Mohit and Contributors, 2026) extends the Jev format to multimodal states. It has 173 dev entries in three subsets. In *visual*, the answer must be read from an image, e.g., the current price of an item on a menu board. In *text*, the state is given entirely in text, e.g., checking an invoice’s lines against its stated total under a rule. In *joint*, a rule stated in text must be applied to values that appear only in the image, e.g., approving an order whose total is computed from the prices on the board.

Multimodal understanding. We cast standard multiple-choice understanding benchmarks as candidate scoring. The question and any image or video form the request, and each answer option is embedded as a candidate.

- **MMLU** (Hendrycks et al., 2021): text-only knowledge-related multi-choice questions. There are 57 subjects in total.
- **MMMU** (Yue et al., 2024): college-level image-text multi-choice questions across 30 subjects.
- **Video-MMMU** (Hu et al., 2025): multi-choice questions about professional lecture videos, organized into 3 tracks (Perception, Comprehension, Adaptation).
- **NaturalBench** (Li et al., 2024): adversarially paired natural image-question examples that penalize answering from language priors alone. We evaluate 4,000 two-option rows.
- **TempCompass** (Liu et al., 2024): fine-grained temporal perception in video. We evaluate its multiple-choice format, using the full option sentence as the candidate rather than the option letter.

Retrieval.

- **NanoBEIR** (AI, 2025): 13 multilingual text retrieval tasks of BEIR (Thakur et al., 2021). Each query is scored against the task’s full passage corpus.

Capability	Benchmark	Modality	Candidates	Tasks	Metric
Decision making	JEVBench	Text	Closed	3	Accuracy
	ImaJEV-Bench	Multimodal	Closed	3	Accuracy
Multimodal understanding	MMLU	Text	Closed	57	Hit@1
	MMMU	Multimodal	Closed	30	Hit@1
	Video-MMMU	Multimodal	Closed	3	Hit@1
	NaturalBench	Multimodal	Closed	1	Hit@1
	TempCompass	Multimodal	Closed	1	Hit@1
Retrieval	NanoBEIR	Text	Open	13	nDCG@10
	MMEB-V3 Image	Multimodal	Open	37	Hit@1
	MMEB-V3 Video	Multimodal	Open	18	Hit@1
	MMEB-V3 VisDoc	Multimodal	Open	24	nDCG@5
Total				190	

Table 1 Evaluation benchmarks. *Tasks* counts the scored task units for each suite. Closed-set tasks score each request against its own option set (fewer than 256 candidates). Open-set tasks score against a shared and larger candidate pool.

- **MMEB-V3** (Huang et al., 2026): the image (37 tasks), video (18 tasks), and visual-document (24 tasks) partitions of the MMEB-V3. They cover multimodal classification, question answering, grounding, and retrieval, all formulated as ranking candidates for a multimodal query.

4.2 Evaluation

All benchmarks are scored the same way. The request is rendered in natural language and encoded once with the last-token representation (Section 3.2), and each candidate is encoded independently. Candidates are then ranked by cosine similarity. **The prompt used for each benchmark is provided in App. A.**

Closed set. Each request is scored only against its own options, and the prediction is the highest-scoring option. This small set of options is added to the request so that the model has a global picture for decision making. We report accuracy, which is Hit@1 over the request’s option set.

Open set. Each request is scored against the full candidate pool of its task. The prompt contains only the state and task description, but not the options. For NanoBEIR we report nDCG@10 with linear gain. For MMEB-V3 we follow the benchmark’s protocol and report Hit@1 on the image and video partitions and nDCG@5 on visual documents.

Results Table 2 reports the results MetaEncoder across all evaluation suites. As baselines we include the SOTA decision making model OpenJev-27B (Wortega and Contributors, 2026), and two SOTA multimodal retrieval models leading the MMEB benchmark, including WeMM-Embedding-9B (Zhou et al., 2026) and Qwen3-VL-Embedding-8B (Zhang et al., 2025b).

The results of MetaEncoder demonstrates that a unified bi-encoder architecture can effectively master both multimodal decision-making and retrieval tasks. On decision-making and multimodal multiple-choice understanding benchmarks, MetaEncoder outperforms current SOTA retrieval models, including WeMM-Embedding-9B, which holds first place on the MMEB-V3 leaderboard (TIGER-Lab, 2026). However, MetaEncoder slightly lags behind OpenJev-27B in text-only decision-making. OpenJev-27B is a SOTA System One model that jointly encodes the request and all options using a single backbone paired with a decision scoring head. We attribute this performance gap to the structural differences between bi-encoders and full-attention models, reflecting the classic tradeoff between bi-encoder caching efficiency and cross-encoder ranking accuracy. However, OpenJev-27B does not have the ability for retrieval tasks nor multimodal tasks.

4.3 Analysis

4.3.1 Whether to Include Options in Prompt

Bi-encoders encode requests and candidates independently, computing scores via inner product instead of cross-attention. This architecture is effective when candidates are semantically distinct, but multiple-

Category	Benchmark	MetaEncoder	OpenJev (27B)	WeMM-Embedding (9B)	Qwen3-VL-Embedding (8B)
Decision making	JevBench (orig)	0.9444	0.9861	0.6250	0.6389
	JevBench (easy)	1.0	1.0	0.9792	0.9375
	JevBench (hard)	0.7387	0.7658	0.4865	0.4324
	Jev Intelligence	81.4	84.5	45.8	42.1
Multimodal decision making	ImaJev (dev)	0.8555	–	0.8117	0.7792
General knowledge understanding	MMLU	0.7767	0.8601	0.7325	0.6544
Multimodal understanding	MMMU	0.5788	–	0.5656	0.5062
	VideoMMMU	0.59	–	0.4846	0.4187
	NaturalBench	0.813	–	0.8017	0.7097
	TempCompass	0.7346	–	0.7527	0.7172
Multilingual text retrieval	NanoBEIR	0.6634	–	0.6175	0.6061
Multimodal retrieval	MMEB V3 (image)	0.7897	–	0.8099	0.7796
	MMEB V3 (video)	0.6486	–	0.7431	0.6715
	MMEB V3 (visdoc)	0.8149	–	0.8334	0.8236

Table 2 Benchmark Evaluation Results for MetaEncoder compared to SOTA System One decision makers and SOTA retrieval models.

Benchmark	Without options	With options	Δ
MMLU (57 subjects)	0.5645	0.7456	+0.1811
MMMU (30 subjects)	0.4875	0.5629	+0.0754
MMMU-Pro (4 options)	0.4699	0.5553	+0.0854
MMMU-Pro (10 options)	0.3503	0.4237	+0.0734

Table 3 Effect of appending candidate option lists to the request prompt. Scores reflect unweighted subject means (MMLU, MMMU) or Hit@1 (MMMU-Pro).

choice benchmarks frequently present options differing by a single attribute—such as "red, oxidative, slow contracting" versus "red, oxidative, fast contracting". To equip the encoder with a global picture of what candidates are available, one solution we adopt is to put the option list directly into the request if the cardinality is small. In this ablation study, we compare evaluation results with and without the options in the prompt. The ablation is conducted on MMLU and MMMU.

Table 3 summarizes the main results, with detailed subject breakdowns provided in Tables 9 and 10. We find that **appending option lists to request yields substantial, uniform gains**: performance improves across 56 of 57 MMLU subjects and 27 of 30 MMMU subjects. These gains are more significant on text-only benchmarks than on multimodal ones.

Recommendation: Always append option list to the request for smaller candidate pool.

4.3.2 Candidate Format in Target

In bi-encoders the candidate string is the entire signal on the candidate side and the only interaction between a request-candidate pair is an inner product between the two vectors. In this section, we measure the impact of the candidate-side format to the final result.

We compare 3 variants of candidate renderings on MMLU and MMMU. In all variants, the request is always the options-in-prompt form (option list and its letters are enumerated in the request); only the candidate strings change:

Rendering	Example
C0 description only	"red, oxidative, slow contracting."
C1 bare label	"C"
C2 label and description	"C. red, oxidative, slow contracting."

The results of the three variants are in Table 4. In summary, including both labels and descriptions (C2)

Benchmark	C0	C1	C2	Δ_{C1}	Δ_{C2}
MMLU (57 subjects)	0.7456	0.6512	0.7767	-0.0944	+0.0311
MMMU (30 subjects)	0.5629	0.4983	0.5788	-0.0646	+0.0159
MMMU-Pro	0.4916	0.3803	0.5042	-0.1113	+0.0126

Table 4 Candidate rendering, with the request held fixed at the options-in-prompt form. C0 description only. C1: bare label. C2: label and description. Hit@1 is the metric over all rows; Δ is against C0.

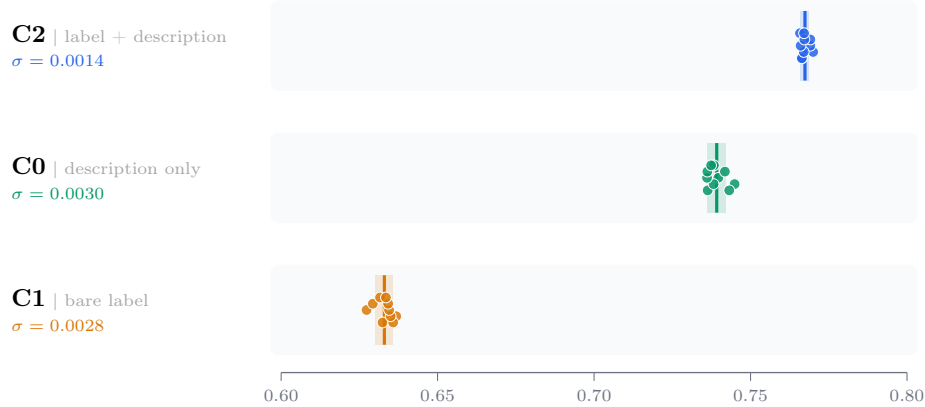


Figure 1 10 random permutations of option order across three variants on MMLU. Standard deviation is measured.

works the best. Meanwhile, using only positional labels (C1) in the candidate side regresses all benchmarks. Despite so, it should be noted that using label only (C1) still reaches 0.6512 on MMLU against a chance floor of 0.2500. This is a clean demonstration that the model resolves a pointer: the letter "C" carries no information about the option semantics and yet it recovers most of the task. However, resolution also degrades as the option set grows: on MMMU the C1 deficit widens monotonically with the number of options, from +0.0281 at three options to -0.0792 at four, -0.1344 at five and -0.1725 at ten, against the C0 baseline. A pointer is harder to resolve more options.

Recommendation: Render each candidate as its identifier and its description.

4.3.3 Sensitivity to candidate order

The previous sections analyze the request and candidate representations which can include labels (A...Z). A natural question to ask is the robustness of option ordering in both the request and candidate sides.

On MMLU, we permute the option order for each candidate which is reflected inside the request and also at the candidate side for C1 and C2. For C0, it affects request only since the candidate is label free. We run 10 independent permutations for that the candidate orders are shuffle randomly in each permutation. We then visualize the score distribution for C0, C1 and C2.

Figure 1 shows the results of all 10 runs. In summary, **candidate order barely matters for all variants**. Across the 10 permutations, none of the three standard deviation exceeds 0.003.

Conclusion The bi-encoder based model is robust to option orders in the prompt and at the target side.

4.3.4 Video understanding

All above analysis focuses on text and image benchmarks. In this section we deep analyze MetaEncoder video understanding on Video MMMU benchmark, focusing on the impact of the number of frames for encoding, as well as the rendering of the request and of the candidates.

Setup. We follow the official Video-MMMU protocol: each of the three tracks (Perception, Comprehension, Adaptation) is scored over its 300 questions and the overall score is their mean. Frames are sampled uniformly from 64 extracted per clip.

Request	Candidate				
	(A) text	A. text	text	(A)	A
Default instruction, (A) list	0.5800	0.5656	0.4933	0.4789	0.4556
Select the correct option., (A) list	0.5711	0.5667	0.5011	0.5178	0.4789
Select the correct option., A. list	0.5544	0.5556	0.4989	0.4100	0.3744

Table 5 Video-MMMU overall accuracy (official protocol, 32 frames) by request and candidate rendering. Rows vary the request: the default instruction (Appendix A.3) or **Select the correct option.**, with options listed as (A) ... or A. Columns vary the candidate.

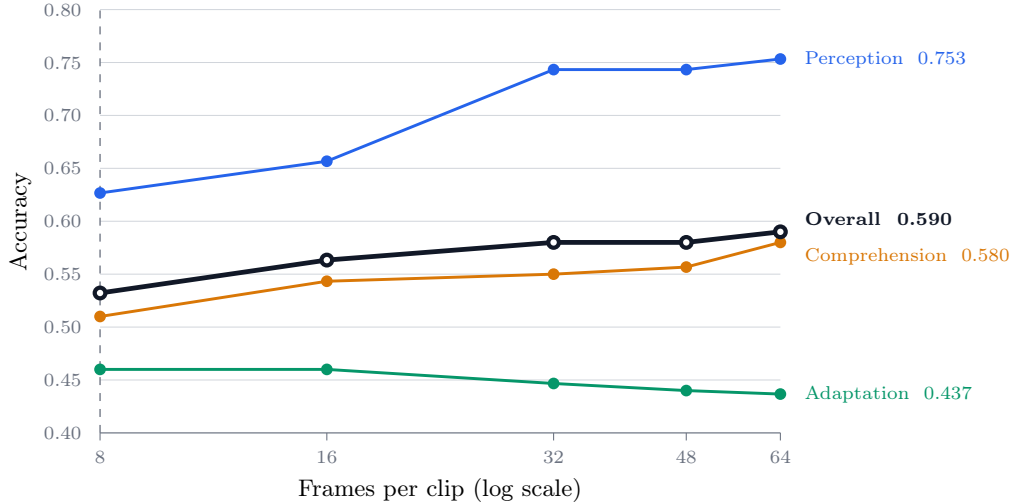


Figure 2 Video-MMMU accuracy as a function of the number of frames sampled per clip, with the request and candidate rendering fixed at the defaults of Appendix A.3. Accuracy follows the official protocol: each track is scored over its 300 questions (open-ended Adaptation questions count as wrong) and *Overall* is the mean of the three tracks. Doubling from 8 to 16 and from 16 to 32 frames adds 0.021 and 0.027 overall.

More frames help. As shown in Figure 2, accuracy rises monotonically with the frame budget, from 0.5322 at 8 frames to 0.5800 at 32 and 0.5900 at 64. Most of the gain arrives by 32 frames. Repeated runs of the same configuration agree within 0.0011 standard deviation, so the trend is well above run-to-run noise. Because video tokens grow linearly with the number of frames, the 64-frame setting costs roughly eight times for a 0.058 gain.

The gain is perceptual. The improvement is concentrated where the answer is on screen. Perception rises by 0.127 (from 0.6267 to 0.7533) and Comprehension by 0.070, while Adaptation falls slightly (0.4600 to 0.4367). By question type, OCR improves from 0.6173 to 0.7509; concept and problem-solving comprehension improve by 0.064 and 0.078. However, the two Adaptation types do not improve (problem-solving adaptation regressed from 0.4071 to 0.4000; case-study analysis regressed from 0.5827 to 0.5396). This indicates that for tasks requiring more complex reasoning, the performance bottleneck of encoder-only architectures becomes more pronounced.

Request and candidate representations also matter for video understanding. Table 5 varies the request and candidate rendering at 32 frames. The pattern matches the closed-set findings of Sections 4.3.1 and 4.3.2. Removing the option list from the request leads to a large regression. With the list present, the best candidate is the option’s letter together with its text. Dropping the letter costs 0.087, because the letter is what binds the candidate to its entry in the request, and dropping the text costs 0.101 to 0.124, because a bare letter carries no content to match.

Conclusion. Increasing frame count enhances video understanding, significantly boosting perception and modestly improving comprehension. However, it yields no benefit for more complex adaptation tasks. Prompt

Benchmark	Decoder	Encoder	Agreement
MMLU	0.7743	0.7740	0.8195
MMMU	0.4962	0.5730	0.7154
MMMU-Pro	0.4441	0.5039	0.7221
4 options	0.4976	0.5601	0.7200
10 options	0.3906	0.4477	0.7242
Video-MMMU	0.5267	0.5415	0.7531
Perception	0.6400	0.6333	-
Comprehension	0.4667	0.5000	-
Adaptation	0.4695	0.4875	-

Table 6 Encoder versus constrained decoder. Video uses 8 frames. *Agreement* denotes the fraction of instances where both models are simultaneously correct or incorrect.

rendering findings transfer directly from text and image domains to video: options should be explicitly included in request, and each candidate should use both its label and full content.

4.3.5 Encoder versus Constrained Decoder

This section compares the MetaEncoder architecture against a constrained decoder baseline. For comparison, we use the pre-trained **Muse-Glimmer** backbone as a decoder, prompting it to generate an option label in a single constrained decoding step. Like the encoder setup, the decoder processes the request, including all candidate options, during prompt pre-filling. It then uses its output projection layer (restricted to valid option tokens A–Z) to emit logit scores for each option. Because the projection head is pruned to these candidate tokens, this approach is as computationally efficient as an encoder with a label projection head. Theoretically, this allows the decoder to evaluate all options jointly using its pre-trained next-token prediction capability while having full attention to request and options.

Table 6 presents the comparison. On MMLU, MetaEncoder and the constrained decoder perform comparably. On MMMU and MMMU-Pro and Video-MMMU, MetaEncoder all outperforms the decoder. We attribute the encoder’s better performance to three factors: 1) MetaEncoder is fine-tuned on contrastive data targeting this exact scoring task, whereas the decoder is trained for general next-token prediction; 2) MetaEncoder incorporates option candidates directly into the prompt, giving it full context parity with the decoder; and 3) constrained single-token decoding deprives the decoder of intermediate chain-of-thought (CoT) reasoning.

4.3.6 Missing Reasoning Ability

To test the third hypothesis above, we measured the total probability mass of **Muse-Glimmer** assigned to valid label tokens prior to the argmax step. Candidate letters account for 0.8679 of the probability mass on MMLU and 0.8434 on Video-MMMU, but only 0.7424 on MMMU, indicating that the decoder is reluctant to emit a direct option letter on complex reasoning tasks without prior step-by-step generation. To confirm this hypothesis, we disabled constrained decoding for the **Muse-Glimmer** decoder, allowing unconstrained generation. As shown in Figure 3, accuracy on MMMU jumped from 0.4962 to 0.7202. This improvement indicates that the primary performance bottleneck lies not in the scoring mechanism itself, but in whether intermediate reasoning is provided. Neither MetaEncoder nor the constrained decoder performs explicit reasoning, resulting in a $\sim 20\%$ gap compared to an unconstrained decoder on MMMU.

As shown in Figure 3, we run further experiments on MMMU to show that providing pre-generated reasoning traces can improve MetaEncoder as well. Prepending CoT reasoning generated by **Muse-Glimmer** to the encoder prompt boosts MetaEncoder’s MMMU accuracy from 0.5730 to 0.7155. Because the generated CoT explicitly reveals the answer label in 93.7% of cases, we also evaluated a setting where final labels were truncated from the CoT to eliminate direct label leakage. Even without label leakage, MetaEncoder reaches 0.6871 accuracy on MMMU. This demonstrates that intermediate reasoning is the primary bottleneck for scoring models regardless of architecture, and incorporating reasoning traces effectively bridges this gap.

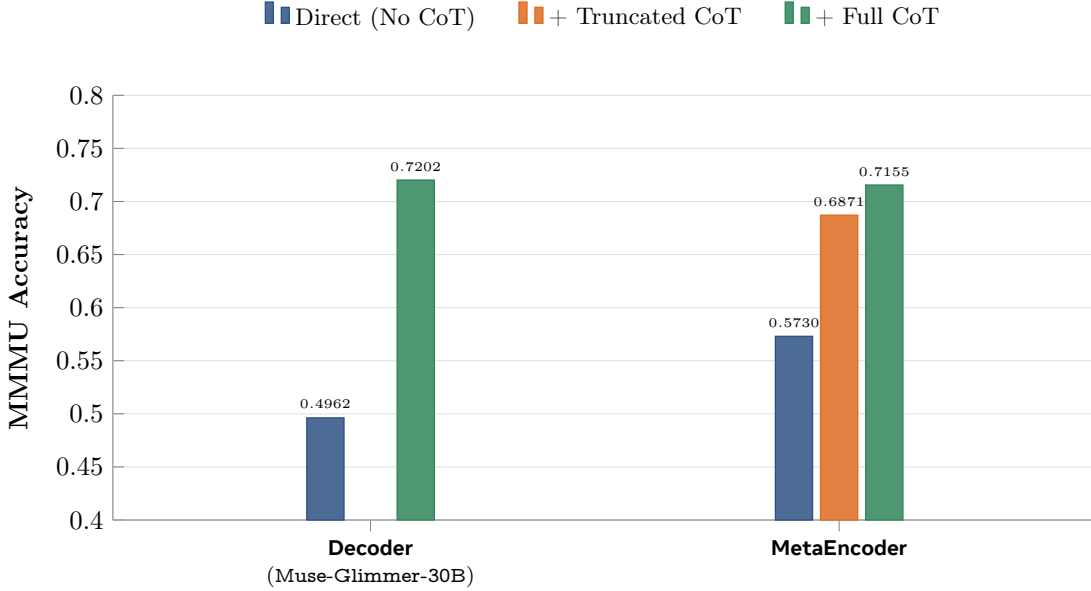


Figure 3 Impact of Chain-of-Thought (CoT) reasoning on MMMU accuracy. Disabling constrained single-token decoding for the Decoder or prepending CoT traces to MetaEncoder substantially bridges the performance gap ($\sim 20\%$).

5 Conclusion

We present MetaEncoder, a multimodal bi-encoder that exposes a single natural language interface for closed-set decision-making, understanding, and open-set retrieval tasks. By utilizing a bi-encoder architecture, MetaEncoder enables efficient representation caching. Furthermore, we demonstrate that appending candidate options directly to the prompt allows bi-encoders to execute decision-making tasks effectively, narrowing the performance gap with cross-encoders and constrained decoders. Comprehensive evaluation across 11 benchmarks spanning multimodal decision-making, understanding, and retrieval demonstrates competitive performance.

Notably, candidate-pool size poses minimal difficulty for MetaEncoder: retrieving from a pool of 1,000 candidates is not harder for MetaEncoder than resolving a four-way multiple-choice question whose options differ by only a single attribute. Instead, task difficulty is dictated by whether an answer possesses clear semantic boundaries or requires pre-retrieval reasoning. Consequently, while MetaEncoder excels at reading, perception, classification, and single-step judgment, it shows weaker performance on complex comprehension and adaptation. To resolve this decision-accuracy bottleneck, reasoning capabilities have to be injected by either leveraging a decoder’s CoT as additional context, or making the encoder model be able to reason directly.

References

- Sionic AI. Nano-beir: A multilingual information retrieval benchmark with quality-enhanced queries, 2025. <https://huggingface.co/sionic-ai>.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *Conference on Language Modeling (COLM)*, 2024.
- Haonan Chen, Chu Li, Zhicheng Wang, Yuanwei Liu, Yuanjiang Wang, Shaohua Jiang, and Zhicheng Dou. Douyin multimodal embedding model technical report. *arXiv preprint arXiv:2608.02148*, 2026.
- Jianpeng Cheng, Xian Wu, Jiangfan Zhang, Wentao Bao, Chaitanya Ahuja, Shlok Kumar Mishra, Hanchao Yu, Yang Gao, Fan Xia, Qi Guo, et al. Tte-flash: Accelerating reasoning-based multimodal representations via think-then-embed tokens. *arXiv preprint arXiv:2605.16638*, 2026.
- Xuanming Cui, Hong-You Chen, Hao Yu, Hao Yuan, Zihao Wang, Shlok Kumar Mishra, Hanchao Yu, Yonghuan Yang, Jun Xiao, Ser-Nam Lim, et al. Reason to contrast: A cascaded multimodal retrieval framework. *arXiv preprint arXiv:2602.23369*, 2025.
- Xuanming Cui, Jianpeng Cheng, Hong-you Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Mishra, Taipeng Tian, Qi Guo, et al. Think then embed: Generative context improves multimodal embedding. In *International Conference on Learning Representations*, volume 2026, pages 2690–2709, 2026.
- Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. FIRST: Faster improved listwise reranking with single token decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8642–8652, 2024.
- Tiancheng Gu, Kaicheng Yang, Kaichen Zhang, Xiang An, Ziyong Feng, Yueyi Zhang, Weidong Cai, Jiankang Deng, and Lidong Bing. Unime-v2: Mllm-as-a-judge for universal multimodal embedding learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 21378–21386, 2026.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.
- Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.
- Haohang Huang, Xuan Lu, Mingyi Su, Xuan Zhang, Ziyang Jiang, Ping Nie, Kai Zou, Tomas Pfister, Wenhu Chen, Wei Zhang, et al. Mmeb-v3: Measuring the performance gaps of omni-modality embedding models. *arXiv preprint arXiv:2604.23321*, 2026.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- Haonan Jiang, Yuji Wang, Yongjie Zhu, Xin Lu, Wenyu Qin, Meng Wang, Pengfei Wan, and Yansong Tang. Embed-rl: Reinforcement learning for reasoning-driven multimodal embeddings. In *European Conference on Computer Vision*, pages 39–56. Springer, 2026.
- Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024.
- Ziyang Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *International Conference on Learning Representations (ICLR)*, 2025.
- Vladimir Karpukhin, Barlas Oğuz, 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)*, pages 6769–6781, 2020.
- Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, 2020.

- Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Ume-r1: Exploring reasoning-driven generative multimodal embeddings. In *International Conference on Learning Representations*, volume 2026, pages 115779–115806, 2026.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *International Conference on Learning Representations (ICLR)*, 2025.
- Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. NaturalBench: Evaluating vision-language models on natural adversarial samples. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibao Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.
- 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.
- Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. Mm-embed: Universal multimodal retrieval with multimodal llms. In *International Conference on Learning Representations*, volume 2025, pages 44215–44234, 2025.
- Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8731–8772, 2024.
- Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning LLaMA for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425, 2024.
- Meta Superintelligence Lab. Muse glimmer-30b model card. <https://huggingface.co/meta-models/Muse-Glimmer-30B>, august 2026. Hugging Face Model Repository.
- Mohit and Contributors. Imajevbench: Benchmark for typed decisions from visual evidence and rules. <https://huggingface.co/datasets/mohit67890/imajev-bench>, september 2026. Dataset and Multimodal Evaluation Suite.
- 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*, pages 2014–2037, 2023.
- Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *International Conference on Learning Representations (ICLR)*, 2025.
- Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.
- Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020.
- Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, 2024.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.
- Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations (ICLR)*, 2023.
- Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, 2021.

- Florian Standhartinger and Contributors. Jevbench: Metamorphic coherence testing for typed probabilistic decision models. <https://github.com/fstandhartinger/jevbench>, september 2026. Benchmark Repository and Evaluation Suite.
- 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*, pages 1102–1121, 2023.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, 2023.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.
- Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5228–5238, 2022.
- TIGER-Lab. Massive multimodal embedding benchmark (mmemb) leaderboard. <https://huggingface.co/spaces/TIGER-Lab/MMEB-Leaderboard>, 2026. Accessed: 2026-10-07.
- TypeSafe AI. Introducing system one models & jev. <https://typesafe.ai>, september 2026. Technical Announcement and Early Access Release.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 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, 2024.
- Alex Wortega and Contributors. Openjev: Qwen3.5-based system one cross-encoder decision model. <https://huggingface.co/openjev/openjev>, 2026. Accessed: 2026-10-07.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, 2024.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruochi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567, 2024.
- Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations (ICLR)*, 2023.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023.
- Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9274–9285, 2025a.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*, 2024.

- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, pages 46595–46623, 2023.
- Junjie Zhou, Ke Mei, Lei Li, Tianyi Wang, Fengyun Rao, and Jing Lyu. Wemm-embedding: Wechat multi-modal embedding technical report. *arXiv preprint arXiv:2608.24053*, 2026.

Appendix

A Prompts

This appendix gives the exact prompt used for every evaluation suite, aligning with MetaEncoder’s natural language interface.

A.1 Common template

All requests and candidates are assembled into a two-turn chat similar to a instruction-following decoder:

```
system: Represent the user’s input.  
user: {request}    (plus any image or video content parts)
```

The system turn is a constant, always applied to both prompt and candidate encodings. The request goes to the user turn, which includes user instructions, context descriptions, criteria etc. Media are attached in the MetaEncoder input interface. The chat template is applied with `add_generation_prompt=False`, so the sequence ends at the user turn’s `<|eot|>`, which is what last-token pooling reads. Candidates are encoded with the same template above, unless noted below, carry only their own text.

A.2 Decision making

JEVBench and ImaJEV-Bench. Each row carries its own task definition and label ontology, rendered by a single shared function:

```
{state}  
Task: {instruction}  
Criteria:  
- {label}: {meaning of the label}  
Options: a, b, c
```

The criteria block is what makes these rows transferable: the label vocabulary is defined in the prompt, so a row whose label set the model has never seen is still answerable. Boolean fields are rendered as `yes: {description}` and `no: {description}` rather than `True` and `False`, since a bare `"True"` is an opaque token that carries no information about what the row is asking.

S1MB (Additional Eval). The instruction is taken from each decision’s `instructions_json` field and the case from `state_json`, rendered as `key: value` lines. Candidates are the natural-language `criteria[].description_json` strings. The source’s criterion identifiers (`"B"`, `"option_3"`, `"0"`) are used only to resolve which candidate is correct and never reach the model. One row is emitted per (case, decision) pair, because a single case may carry several decisions with different answer spaces.

An agentic control example:

```
Choose one permitted tactical maneuver using the measured scene. The simulator retains its collision  
reflex and flight controller.  
  
adapter: drone  
observation: {"control_policy": {"braking_deceleration_mps2": 3, ...}}
```

with candidates such as `"climb: Climb over an obstacle within verified clearance"` and `"gap_right: Use the observed safe gap on the right"`.

For ordinal (`score`) decisions each criterion additionally carries a numeric `value`, which is retained in a sidecar so that predictions can receive graded credit:

```
constraint: Do not infer additional numbers from associations  
question: How much of the claim does the source support?  
unit: Explicit numerical claims
```

claim: Revenue rose 20% and profit rose 20%
source: Revenue rose 20%. Profit figures were not disclosed.

with candidates "Source mentions neither revenue nor profit" (value 0), "Mentions revenue or profit but confirms neither claimed number" (1), "Confirms exactly one claimed number" (2) and "Confirms both claimed numbers" (3).

A.3 Multimodal understanding

Video-MMMU and TempCompass. These three share one template, with the options rendered into the request:

Given a video and a question, select the most accurate answer from the provided candidates. Return only the exact text of your chosen answer. Question:{question}
Options:
(A) {option}
(B) {option} ...

Candidates are the full option sentence together with its letter marker, for example "(A) To get her to trust him". A verbatim MVBench request:

Given a video and a question, select the most accurate answer from the provided candidates. Return only the exact text of your chosen answer. Question:Why did Castle dress like a fairy when he was speaking to Emily?
Options:
(A) To get her to trust him
(B) He secretly loved fairies ...

TempCompass Options arrive letter-prefixed in the source ("B. A man is dunking a basketball."). The gold is resolved by matching that prefix, after which the prefix is stripped, so the candidate is the sentence alone. Video inputs use 8 uniformly sampled frames drawn from 64 extracted per clip.

Video-MMMU *Adaptation* track additionally references a still image. Frames are passed through a separate video channel and the still as its own image content part, so that both reach the model, and the `<image 1>` placeholder in the question text is reduced to the word `image`.

MMLU and MMMU. The base request is:

Answer the multiple-choice question by selecting the correct option.
question: {question}

In the options-in-prompt variant, marked * in the results tables, the alternatives are additionally rendered into the request:

Options:
A. {option}
B. {option} ...

Candidates are the full option text in both variants, never the letter. MMMU's `<image N>` placeholders are reduced to the word `image`, since the referenced images travel as content parts; rows whose candidates consist *only* of an image reference are excluded, because such a candidate carries no matchable text. A verbatim MMLU* request:

Answer the multiple-choice question by selecting the correct option.
question: Type I muscle fibres have the following characteristics:
Options:
A. white, glycolytic, slow contracting.
B. red, oxidative, fast contracting.
C. red, oxidative, slow contracting.
D. white, oxidative, slow contracting.

NaturalBench. The dataset's own yes/no question forms the request, with the image attached as a content part; the candidates are **Yes** and **No**. Each group pairs two images with two questions whose answers flip between the images, so answering from the question alone cannot succeed on all four cells.

A.4 Retrieval

NanoBEIR. Queries carry the task instruction for their subset; passages are encoded without an instruction. The instructions are, verbatim:

Subset	Query instruction
NanoMSMARCO	Given a web search query, retrieve relevant passages that answer the query
NanoNQ	Given a question, retrieve passages that answer the question
NanoHotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
NanoFEVER	Given a claim, retrieve documents that support or refute the claim
NanoNFCorpus	Given a question, retrieve relevant documents that best answer the question
NanoFiQA2018	Given a financial question, retrieve user replies that best answer the question
NanoQuoraRetrieval	Given a question, retrieve questions that are semantically equivalent
NanoArguAna	Given a claim, find documents that refute the claim
NanoClimateFEVER	Given a claim about climate change, retrieve documents that support or refute the claim
NanoDBPedia	Given a query, retrieve relevant entity descriptions from DBPedia
NanoSCIDOCS	Given a scientific paper title, retrieve paper abstracts that are cited by the given paper
NanoSciFact	Given a scientific claim, retrieve documents that support or refute the claim
NanoTouche2020	Given a question, retrieve detailed and persuasive arguments that answer the question

MMEB-V3. For MMEB-V3 each task ships its own query instruction in the benchmark metadata, and we use it directly in prompt. The request is assembled as `qry_instuction` followed by the row’s `qry_text`, with the image attached as a content part; candidates are built from `tgt_text` and `tgt_img_path`. Instructions are per task, and their form follows the task type: classification tasks ask for a representation of the image, retrieval tasks describe what should be retrieved, and grounding tasks describe the region to select.

Task	qry_inst
ImageNet-1K	Represent the given image for classification
Country211	Identify the country depicted in the image
DocVQA	Represent the given image with the following question:
RefCOCO	Select the portion of the image that follows the language expressions.
OVEN	Retrieve a Wikipedia image-description pair that provides evidence for the question of this image:
FashionIQ	Find an image to match the fashion image and style note:
MSCOCO_t2i	Find me an everyday image that matches the given caption:
VisDial	Represent the given dialogue about an image, which is used for image retrieval:

Task	Type	Acc
open_jev_browser_control_v1	choice	1
laya_enron_spam	noul	0.96
miqa	choice	0.96
open_jev_amount_extraction_control_v1	noul	0.96
open_jev_mailroom_control_v1	noul	0.95
winobias	choice	0.94
followir_robust04	noul	0.935
workflow_controls_customer_service	noul	0.93
generalization_contextual_choice	choice	0.93
open_jev_silent_failure_control_v1	noul	0.92
hans	noul	0.91
go_emotions	noul	0.9085
workflow_controls_invoice_processing	noul	0.9
workflow_controls_security_incidents	noul	0.9
hatecheck	noul	0.89
impli	noul	0.89
workflow_controls_agent_trace_observability	noul	0.89
laya_ag_news	choice	0.88
sdoh_nli	noul	0.88
open_jev_painting_geometry_v1	noul	0.87
generalization_diverse_choice	choice	0.87
defeasible_nli	choice	0.86
open_jev_context_retention_control_v1	noul	0.86
creak	noul	0.85
dbpedia	choice	0.85
quartz	choice	0.85
snli	choice	0.85
synth_rel_NanoFEVER	score	0.84
synth_rel_NanoSciFact	score	0.8387
paws	noul	0.83
scitail	noul	0.83
arc	choice	0.81
babi_nli	noul	0.81
open_jev_customer_control_v1	score	0.81
open_jev_vizdoom_basic_v1	score	0.803
synth_rel_NanoNQ	score	0.8025
synth_rel_nanocoir	score	0.8
clinc	choice	0.7748
ethos	noul	0.77
open_jev_phone_extraction_control_v1	choice	0.77
cladder	noul	0.76
followir_news21	noul	0.76
synth_rel_NanoHotpotQA	score	0.755
lonli	choice	0.75
gsm8k	choice	0.74
open_jev_ir_control_v1	score	0.74
massive	choice	0.735
openbookqa	choice	0.73
synth_rel_NanoSCIDOCS	score	0.7288
synth_rel_NanoNFCorpus	score	0.7275
followir_core17	noul	0.725
few_nerd	choice	0.7235
civil_comments	noul	0.7229
hwu64	choice	0.72
mtop	choice	0.715
canttalk	noul	0.71
generalization_diverse_noul	noul	0.71
aegis2	noul	0.7027
synth_rel_NanoMSMARCO	score	0.6925
open_jev_sponsor_segment_control_v1	choice	0.69
synth_rel_NanoClimateFEVER	score	0.69
synth_rel_NanoArguAna	score	0.6725
open_jev_citation_control_v1	choice	0.67
open_jev_email_selection_control_v1	noul	0.67
synth_rel_NanoQuoraRetrieval	score	0.6687
arct	choice	0.66
open_jev_entity_alignment_control_v1	noul	0.66
generalization_contextual_noul	noul	0.66
synth_rel_NanoDBPedia	score	0.6462
open_jev_reasoning_control_v1	score	0.64
ruletaker	noul	0.64
scone	choice	0.64
ethics	noul	0.63
synth_rel_NanoFiQA2018	score	0.6275
lexcomp	noul	0.62
plane	noul	0.59
aqua_rat	choice	0.58
bbq	choice	0.58
scicite	choice	0.58
crows_pairs	choice	0.57
robust_lr	choice	0.57
contract_nli	choice	0.5676
gretel_pii	choice	0.5666
banking77	choice	0.56
open_jev_snake_v1	noul	0.56
laya_typed_decisions	score	0.556
logical_entailment	noul	0.55
generalization_diverse_score	score	0.55
temporal_nli	choice	0.55
generalization_contextual_score	score	0.53
tracie	noul	0.53
hh_rlhf	choice	0.52
winowhy	noul	0.52
sgd	choice	0.51
nlsat	noul	0.5
ud_ewt	choice	0.4944
qasper	noul	0.49
synth_rel_NanoTouche2020	score	0.4809
boardgameqa	choice	0.48
open_jev_drone_control_v1	noul	0.44
fol_nli	choice	0.43

Task	Type	Acc
wiqa	choice	0.42
corr2cause	choice	0.41
open_jev_tic_tac_toe_v1	choice	0.38
spartqa	choice	0.37
patent_similarity	score	0.36
esci	choice	0.33
poem_sentiment	choice	0.33
stepgame	choice	0.28
argument_quality	noul	0.27

Table 7 Full results on S1MB (110 tasks).

Task	Score
scene_transition	0.925
unexpected_action	0.775
moving_attribute	0.755
object_existence	0.695
character_order	0.675
action_prediction	0.645
object_interaction	0.615
action_sequence	0.585
state_change	0.565
fine_grained_pose	0.56
action_antonym	0.55
fine_grained_action	0.52
moving_direction	0.48
episodic_reasoning	0.465
action_localization	0.425
counterfactual_inference	0.395
object_shuffle	0.375
action_count	0.37
moving_count	0.35
egocentric_navigation	0.35

Table 8 Full results on MVBench (20 tasks).

Subject	Score (options not in prompt)	Score (options in prompt)
security_studies	0.2939	0.702
international_law	0.405	0.7769
human_sexuality	0.4733	0.8168
nutrition	0.4379	0.7778
moral_disputes	0.4335	0.7139
jurisprudence	0.4907	0.7685
high_school_microeconomics	0.5672	0.8361
sociology	0.597	0.8607
high_school_physics	0.4702	0.7285
college_biology	0.6042	0.8542
econometrics	0.4035	0.6404
high_school_chemistry	0.4483	0.6798
high_school_government_and_politics	0.6788	0.9067
clinical_knowledge	0.5434	0.766
high_school_statistics	0.5417	0.7639
computer_security	0.56	0.78
professional_accounting	0.4823	0.7021
high_school_geography	0.6717	0.8889
astronomy	0.5921	0.8092
high_school_macroconomics	0.541	0.7564
electrical_engineering	0.4621	0.6759
us_foreign_policy	0.58	0.79
high_school_computer_science	0.63	0.84
high_school_world_history	0.654	0.8565
abstract_algebra	0.5	0.7
philosophy	0.4952	0.6945
high_school_biology	0.6677	0.8645
high_school_us_history	0.7059	0.902
high_school_european_history	0.6061	0.8
college_mathematics	0.45	0.64
professional_law	0.3937	0.5815
professional_psychology	0.5474	0.732
management	0.6408	0.8252
public_relations	0.5273	0.7091
medical_genetics	0.7	0.88
college_medicine	0.5549	0.7283
prehistory	0.608	0.7685
college_computer_science	0.53	0.69
marketing	0.7479	0.906
high_school_psychology	0.7358	0.8936
conceptual_physics	0.6298	0.766
human_aging	0.5605	0.6951
professional_medicine	0.7132	0.8456
college_physics	0.6765	0.8039
machine_learning	0.5536	0.6786
moral_scenarios	0.3676	0.4883
college_chemistry	0.53	0.65
formal_logic	0.4206	0.5397
miscellaneous	0.7893	0.908
virology	0.4036	0.5181
high_school_mathematics	0.5444	0.6556
anatomy	0.5778	0.6889
logical_fallacies	0.7362	0.8466
world_religions	0.7544	0.848
business_ethics	0.61	0.7
elementary_mathematics	0.7672	0.8439
global_facts	0.57	0.5

Subject	Score (options not in prompt)	Score (options in prompt)
---------	-------------------------------	---------------------------

Table 9 Full results for MMLU (57 tasks).

Subset	Score (options not in prompt)	Score (options in prompt)
Electronics	0.302	0.5302
History	0.5292	0.7468
Psychology	0.497	0.7134
Design	0.6683	0.8492
Manage	0.3613	0.5393
Sociology	0.4769	0.6299
Math	0.3671	0.4968
Chemistry	0.4326	0.5603
Diagnostics and Laboratory Medicine	0.3229	0.4427
Agriculture	0.4146	0.5316
Art Theory	0.6419	0.7533
Biology	0.474	0.5836
Physics	0.4051	0.5118
Materials	0.3627	0.4481
Clinical Medicine	0.5847	0.6695
Geography	0.3811	0.4656
Marketing	0.625	0.7067
Basic Medical Science	0.5347	0.6152
Mechanical Engineering	0.3296	0.4081
Pharmacy	0.5418	0.6058
Energy and Power	0.3764	0.4245
Economics	0.5274	0.5719
Art	0.7395	0.7816
Public Health	0.592	0.6315
Literature	0.8239	0.8551
Computer Science	0.4803	0.5046
Music	0.3444	0.365
Architecture and Engineering	0.3754	0.3483
Finance	0.5655	0.5329
Accounting	0.5486	0.495
Finance	0.5655	0.5329
Accounting	0.5486	0.495

Table 10 Full results for MMMU (30 tasks).

Suite	Dataset	Score
image	EDIS	0.98
image	Visual7W-Pointing	0.972
image	DocVQA	0.965
image	RefCOCO	0.969
image	RefCOCO-Matching	0.942
image	ImageNet-R	0.943
image	TextVQA	0.913
image	WebQA	0.904
image	InfographicsVQA	0.882
image	VisDial	0.874
image	ObjectNet	0.857
image	VOC2007	0.855
image	ImageNet-1K	0.85
image	Wiki-SS-NQ	0.841
image	VisualNews i2t	0.856
image	ChartQA	0.837
image	OVEN	0.836
image	SUN397	0.816
image	VisualNews t2i	0.839
image	N24News	0.816
image	HatefulMemes	0.794
image	MSCOCO t2i	0.801
image	MSCOCO i2t	0.78
image	ScienceQA	0.745
image	OK-VQA	0.757
image	GQA	0.756
image	ImageNet-A	0.721
image	MSCOCO	0.732
image	CIRR	0.685
image	VizWiz	0.674
image	A-OKVQA	0.665
image	NIGHTS	0.672
image	Visual7W	0.592
image	MCMR	0.6685
image	Place365	0.525
image	Country211	0.477
image	FashionIQ	0.427
video	UCF101	0.941
video	ActivityNetQA	0.799
video	SmthSmthV2	0.8
video	NExTQA	0.7477
video	HMDB51	0.716
video	K700	0.718
video	MSVD	0.7239
video	QVHighlight	0.7156
video	DiDeMo	0.5916
video	MVBench	0.5533
video	Video-MME	0.5485
video	MSR-VTT	0.526
video	EgoSchema	0.522
video	VATEX	0.5013
video	MomentSeeker	0.4744
video	Breakfast	0.4665
video	YouCook2	0.2693

Suite	Dataset	Score
video	Charades-STA	0.2682
visdoc	ViDoSeek-doc	0.9974
visdoc	ViDoRe syntheticDocQA artificial intelligence	0.94
visdoc	VisRAG SlideVQA	0.9514
visdoc	ViDoRe tabfquad	0.8964
visdoc	ViDoRe syntheticDocQA energy	0.88
visdoc	ViDoRe syntheticDocQA healthcare industry	0.9
visdoc	VisRAG InfoVQA	0.883
visdoc	ViDoRe infovqa	0.8583
visdoc	ViDoRe syntheticDocQA government reports	0.84
visdoc	ViDoRe arxivqa	0.842
visdoc	VisRAG ArxivQA	0.8333
visdoc	VisRAG MP-DocVQA	0.8359
visdoc	VisRAG ChartQA	0.8571
visdoc	MMLongBench-doc	0.7554
visdoc	ViDoSeek-page	0.7145
visdoc	ViDoRe shiftproject	0.65
visdoc	ViDoRe biomedical lectures v2 multilingual	0.6234
visdoc	ViDoRe esg reports v2 multilingual	0.6053
visdoc	VisRAG PlotQA	0.5817
visdoc	ViDoRe esg reports human labeled v2	0.6731
visdoc	ViDoRe tatdqa	0.5765
visdoc	ViDoRe economics reports v2 multilingual	0.6422
visdoc	MMLongBench-page	0.5215
visdoc	ViDoRe docvqa	0.5366

Table 11 Full results on MMEB-V3 image, video and visdoc (79 tasks).

Task	NDCG@10
Touche2020 *	0.4609
NQ	0.8311
QuoraRetrieval	0.949
ClimateFEVER	0.2869
FiQA2018	0.7123
MSMARCO	0.5819
DBPedia *	0.6235
ArguAna	0.6537
FEVER	0.9534
NFCorpus	0.3789
HotpotQA *	0.8943
SCIDOCS *	0.4885
SciFact	0.8075

Table 12 Full results on Nano-BEIR (13 tasks).