COMPRESS AT INPUT, RECOVER AT DEPTH: IMPROVING THE PERFORMANCE–EFFICIENCY TRADE-OFF FOR MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In common MLLM pipelines, full-token connectors (e.g., Linear or MLP projectors) forward all modality tokens into the decoder, making attention cost grow quadratically with input length. Token compression alleviates this cost but can weaken multimodal conditioning, creating a core efficiency–performance trade-off. This issue is particularly pronounced for long or information-sparse scientific modalities, where aggressive compression can discard task-relevant input information. We propose **ENCORE**, a dual-stage connector built on a simple principle: *compress at input, recover at depth*. ENCORE constructs a compact set of instruction-conditioned query tokens to reduce the number of input modality tokens, and introduces a lightweight one-shot recovery route that uses intermediate text states to retrieve complementary information from pre-compression modality memory. A stabilized residual update integrates this information during prefill, with no repeated memory reads during cached decoding. Across vision, protein, and molecule tasks, ENCORE achieves the best overall Norm-Avg against the shared MLP and Q-Former references while improving the performance–efficiency trade-off. Relative to MLP, it reduces inference FLOPs by 16.3–62.6% and lowers training time in four of five settings. Extended comparisons further show that ENCORE remains competitive with stronger compression and multi-layer reinjection baselines while retaining a compact decoder interface. Ablations show that compression provides a strong primary representation, while one-shot recovery contributes complementary gains, especially under aggressive compression. Overall, ENCORE provides a practical, modality-agnostic connector that consistently improves the performance–efficiency trade-off across diverse settings.

1 INTRODUCTION

Multimodal large language models (MLLMs) can be built in many ways, but the most widely adopted approach connects a pretrained modality encoder to an LLM through a lightweight connector (Zhu et al., 2023; Bai et al., 2023; Li et al., 2023a). This modular pipeline has been adopted in vision–language systems (Liu et al., 2023a; 2024a) and, due to its minimal architectural requirements, extends naturally to scientific modalities such as proteins and small molecules (Xiao et al., 2025; Wang et al., 2026b; Liang et al., 2024). Because both the encoder and the LLM are typically reused from pretrained checkpoints with limited modification, the connector becomes the primary design variable governing how multimodal signals enter the decoder and persist across layers.

A central challenge is that existing connectors trade off information preservation against computational efficiency. Full-token projectors (e.g., Linear or MLP) pass all encoder outputs into the decoder, retaining every encoder token at the input interface. However, this leads to quadratic attention cost in the input length (Vaswani et al., 2017), which is particularly prohibitive for modalities with long sequences, such as proteins (Xiao et al., 2025; Wang et al., 2026b). While post-hoc token pruning and merging can partially mitigate this cost at inference time (Chen et al., 2024; Shang et al., 2026), such methods can degrade downstream performance when they introduce a mismatch between training and inference inputs. By contrast, Q-Former-style architectures compress encoder outputs into a fixed set of latent representations during training (Li et al., 2023a; Alayrac et al., 2022). While this avoids the train–test mismatch, the compression introduces an input information bottleneck that can weaken fine-grained signals and limit their accessibility in later layers (Yang et al., 2025; Cha et al., 2024).

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.

Meanwhile, a separate line of work explores decoder-side multimodal fusion through cross-attention, gated fusion, and residual reinjection (Alayrac et al., 2022; Gao et al., 2023; Liu et al., 2026; Meng et al., 2024). These methods provide richer modality access at depth, but often incur substantial computational overhead through repeated decoder-side interaction (Chen et al., 2026). This raises a distinct design question: can a compact input interface retain its efficiency while selectively recovering complementary modality information at depth?

We address this question with **ENCORE**, a dual-stage connector built on the principle *compress at input, recover at depth*. ENCORE separates the two roles: instruction-conditioned compression produces a short modality sequence that serves as the decoder's primary representation, while a single intermediate-layer recovery route uses text states to retrieve complementary information from pre-compression modality memory. A stabilized residual update integrates the retrieved information into the text stream. The compact input reduces computation throughout the decoder, while recovery provides additional modality access only once during prefill.

To demonstrate its generalizability, we evaluate ENCORE across three modality families—vision, proteins, and molecules—covering 10 benchmark suites/tasks and five LLM backbones. Across these settings, ENCORE consistently improves the performance–efficiency trade-off over full-token, compression-based, and multi-layer reinjection connectors. Compared with full-token MLP, ENCORE reduces inference FLOPs by 16.3–62.6% and lowers training time in four of five settings, with particularly pronounced gains on proteins, where it cuts inference FLOPs by 59.2% and saves up to 41.8 h of training time while improving overall performance. Against recent compression methods, ENCORE achieves higher aggregate performance at similar or lower inference cost, while matching multi-layer reinjection methods at substantially lower FLOPs. Controlled ablations further show that both stages matter: compression provides a strong primary representation, while recovery contributes larger gains as compression becomes more aggressive. Our main contributions are:

- We introduce **ENCORE**, a dual-stage connector that pairs efficient input compression with a single recovery read from pre-compression memory, restoring access to complementary information lost at the compressed interface.
- We show that ENCORE consistently improves the performance–efficiency trade-off across **vision, protein, and molecule** tasks and five LLM backbones, outperforming recent compression methods and matching multi-layer reinjection at substantially lower inference cost.
- We find that compression and recovery play distinct roles: compression forms a strong primary modality representation, while recovery restores sample-specific complementary information at depth, with larger gains for reasoning and under stronger compression.

2 METHOD

We introduce ENCORE, a two-stage connector framework with (i) an embedding-level compression module that produces a compact modality summary M_q, and (ii) a single-layer recovery route that retrieves complementary information from pre-compression modality memory at a designated decoder layer ℓ_r. The first stage shortens the multimodal input to the LLM, while the second restores information beyond the compressed summary through a stabilized residual update.

2.1 PRELIMINARIES

We consider an MLLM with a pretrained modality encoder $\mathcal{E}$ and a decoder backbone $\mathcal{F}$. Given modality input x_m, the encoder produces raw features $V = \mathcal{E}(x_m), V \in \mathbb{R}^{N_v \times d_v}$, where N_v is the raw modality token count and d_v is the encoder feature dimension. From V, we derive a modality summary $M_q \in \mathbb{R}^{N_q \times d}$, where $N_q \ll N_v$, via the instruction-conditioned extractor defined in Sec. 2.2, and a pre-compression modality memory

$$V_m = VW_m, \qquad W_m \in \mathbb{R}^{d_v \times d}, \qquad V_m \in \mathbb{R}^{N_v \times d}, \tag{1}$$

where d denotes the LLM hidden dimension and W_m is an learnable projection used only for recovery. The memory retains all N_v encoder positions for the recovery read. Given text tokens $t_1, \ldots, t_{T_t}$ with T_t denoting the text token count, let $\mathrm{Emb}(\cdot)$ be the token embedding function. The decoder

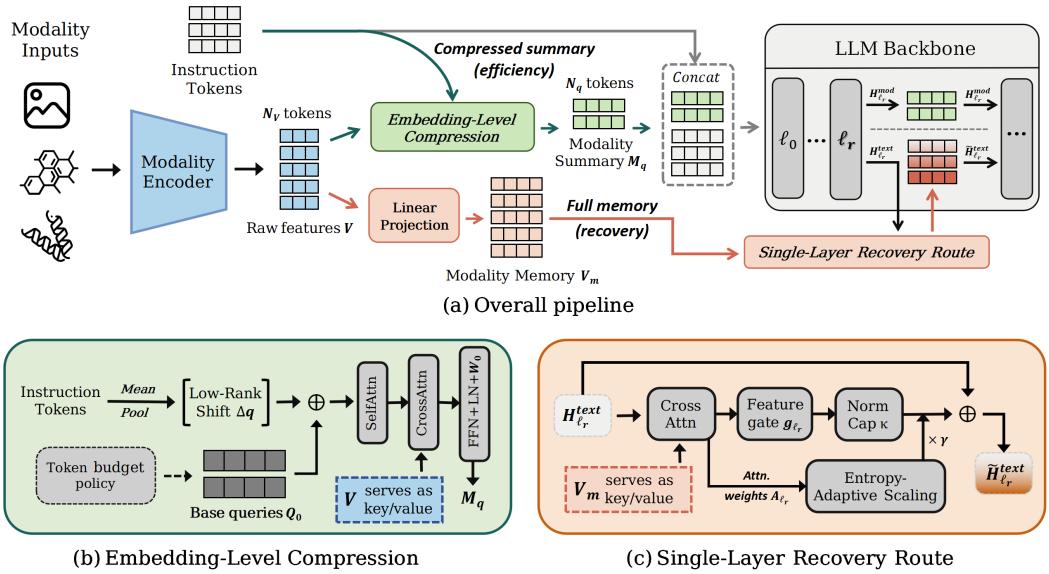

Figure 1: Architecture of ENCORE. (a) The modality features V are compressed into a compact summary M_q ($N_q \ll N_v$), while the full projected memory V_m is retained for single-layer recovery at decoder layer ℓ_r. (b) Input compression with instruction-conditioned queries. (c) Recovery via cross-attention from text states to V_m, followed by stabilized residual integration.

input is formed by concatenating the compressed modality summary with the text embeddings: $H_0 = [\, M_q;\ \mathrm{Emb}(t_1), \ldots, \mathrm{Emb}(t_{T_t})\,]$. We denote the modality-summary and text portions by

$$H_l = [\, H_l^{\mathrm{mod}};\ H_l^{\mathrm{text}}\,], \quad l = 1, \ldots, L, \tag{2}$$

where L is the total number of decoder layers, $H_l^{\mathrm{mod}} \in \mathbb{R}^{N_q \times d}$ corresponds to the modality summary positions, and $H_l^{\mathrm{text}} \in \mathbb{R}^{T_t \times d}$ denotes the text hidden states. We further denote by $\mathcal{T}_{\mathrm{prompt}} \subseteq \{1, \ldots, T_t\}$ the index set of prompt tokens available at inference time (excluding teacher-forced targets), and by $\ell_r \in \{1, \ldots, L\}$ the recovery layer index, with default $\ell_r = \lfloor L/2 \rfloor$ (Sec. 2.3).

Hyperparameter disclosure. ENCORE shares most hyperparameters across modalities and backbones; the remaining settings are determined from modality or backbone properties (App. A.3).

2.2 Embedding-Level Compression

Motivation. Task-agnostic compression summarizes modality features independently of the instruction, which can waste limited query capacity on irrelevant content. We therefore condition the queries on the instruction to focus compression on task-relevant evidence.

Instruction-conditioned query modulation. Let $Q_0 \in \mathbb{R}^{N_q \times d_v}$ denote learnable base queries. Rather than generating prompt-specific queries, we keep the query slots fixed and use the instruction only to modulate them. This preserves a stable slot structure and makes the extractor independent of prompt length. We mean-pool only prompt-token embeddings, excluding teacher-forced targets:

$$s = \mathrm{MeanPool}\big(\{\mathrm{Emb}(t_i) \mid i \in \mathcal{T}_{\mathrm{prompt}}\}\big), \qquad s \in \mathbb{R}^d, \tag{3}$$

where $\mathcal{T}_{\mathrm{prompt}}$ indexes inference-available prompt tokens. A low-rank shift is then computed and broadcast to all query slots:

$$Q = Q_0 + \frac{\mathbf{1}_{N_q}\big((sW_A)W_B\big)}{\sqrt{r_q}}, \qquad W_A \in \mathbb{R}^{d \times r_q}, \qquad W_B \in \mathbb{R}^{r_q \times d_v}, \tag{4}$$

where r_q is the bottleneck rank and $\mathbf{1}_{N_q} \in \mathbb{R}^{N_q \times 1}$ expands the resulting d_v-dimensional shift uniformly across all query slots. This treats the instruction as a global conditioning signal, injecting a shared bias into all queries while preserving slot-level specialization through Q_0.

Lightweight cross-attention extractor. Our extractor follows the query-based compression paradigm of Q-Former (Li et al., 2023a), where a small set of learnable queries cross-attends to modality features to produce a fixed-size summary. Unlike standard Q-Former designs, ENCORE uses instruction-conditioned queries and replaces the deep extractor stack with a single shallow block to keep the input interface lightweight. This compressed summary provides the primary modality representation throughout the decoder; the recovery route supplements it at depth (Sec. 2.3). Concretely,

$$Q_v = \mathrm{FFN}\Big(\mathrm{CrossAttn}\big(\mathrm{SelfAttn}(Q),\, V\big)\Big), \tag{5}$$

$$M_q = \mathrm{LN}(Q_v)\, W_o, \qquad W_o \in \mathbb{R}^{d_v \times d}. \tag{6}$$

Token budget allocation. We balance information retention and attention cost by rounding N_v down to the nearest power of two, scaling by 2^{-k}, and enforcing a minimum budget $N_{\min}$:

$$N_q = \max\left(N_{\min},\ 2^{\lfloor \log_2 N_v \rfloor - k}\right), \tag{7}$$

where k controls the compression ratio and $N_{\min}$ avoids overly small token budgets. We use $k = 2$ and $N_{\min} = 8$, with fixed evaluated budgets of 128, 64, and 8 for vision, protein, and molecule, respectively (Table 5). The effect of the query budget is examined in Sec. 3.3.1.

Efficiency gain from token reduction. Let $N_{\mathrm{base}} = T_t + N_v$ and $N_{\mathrm{enc}} = T_t + N_q$ denote the decoder sequence lengths before and after compression, and ΔN counts the removed modality tokens. Compression reduces the quadratic attention cost from $O(LN_{\mathrm{base}}^2 d)$ to $O(LN_{\mathrm{enc}}^2 d)$, with relative savings depending on both text length and modality token reduction. The per-layer prefill saving is

$$\Delta f_{\mathrm{fwd}} = 24\,\Delta N\, d^2 + 4\,\Delta N\,(2T_t + N_v + N_q)\, d, \qquad \Delta N = N_v - N_q, \tag{8}$$

Across L decoder layers, savings total $L \cdot \Delta f_{\mathrm{fwd}}$, while the extractor and recovery route incur only a one-time overhead. For $N_v > N_q$, savings exceed this overhead beyond a break-even depth L^*. At fixed connector overhead, larger ΔN lowers L^*; for short modality sequences, overhead can limit the gain (derivation in App. C.2; empirical validation in Sec. 3.2.4).

Information compression bottleneck. To isolate the representational cost of token reduction, we compare compression with a token-preserving MLP. For a fixed instruction, let $g_{\mathrm{cmp}} \colon \mathbb{R}^{N_v d_v} \to \mathbb{R}^{N_q d}$ map flattened raw features to the compressed summary, and let $g_{\mathrm{mlp}} \colon \mathbb{R}^{N_v d_v} \to \mathbb{R}^{N_v d}$ denote the reference that retains all N_v tokens. Here, N_v, d_v are the encoder token count and feature dimension, and N_q, d are the summary token count and LLM hidden dimension. For any smooth map g, $J_g(x)$ denotes its Jacobian at x and $\ker(J_g(x))$ its null space.

Proposition 1 (Information loss and null-space gap)**.** *Let $x = \mathrm{vec}(V) \in \mathbb{R}^{N_v d_v}$ be the row-wise vectorization of V, $M_q = g_{\mathrm{cmp}}(x)$, and let $I(\cdot;\cdot)$ denote mutual information.*

(i) ***Information bound.*** *For any deterministic post-processing f, $I(x;\, f(M_q)) \le I(x;\, M_q)$.*

(ii) ***Null-space gap.*** *If g_{mlp} operates at full output rank (i.e., the token-preserving reference does not itself collapse output directions), then g_{cmp} introduces, in the sense of the local Jacobian, at least $(N_v - N_q)\, d$ additional null-space dimensions compared to g_{mlp}.*

The full-output-rank condition requires $d_v \ge d$ and need not hold for every backbone; App. C.3.2 gives the general rank bound. Neither result alone implies task degradation, which depends on whether compression removes distinctions needed by the task.

Implication. By (i), processing M_q alone cannot restore discarded information. By the chain rule, any differentiable module reading only M_q also remains first-order insensitive to directions in $\ker(J_{g_{\mathrm{cmp}}}(x))$. When discarded distinctions are task-relevant (empirical evidence in App. C.3.3), recovery therefore requires access beyond M_q. ENCORE provides this access by bypassing g_{cmp} and re-accessing raw features, as formalized in App. C.3.5.

2.3 Single-Layer Recovery and Integration

Motivation. ENCORE recovers complementary information beyond M_q through a single cross-attention read from the text stream to the pre-compression memory V_m.

Mid-layer recovery. Recovery should occur after text states have processed both the instruction and M_q, while leaving enough decoder depth to integrate the retrieved information (Gupta et al., 2026; Song et al., 2026; Lin et al., 2025). We therefore use the midpoint $\ell_r = \lfloor L/2 \rfloor$ as a shared default across settings. At this layer, the intermediate text states query V_m:

$$Z_{\ell_r} = \mathrm{CrossAttn}\big(H^{\mathrm{text}}_{\ell_r},\, V_m\big), \qquad Z_{\ell_r} \in \mathbb{R}^{T_t \times d}, \tag{9}$$

with head-averaged row-stochastic attention matrix $A_{\ell_r} \in \mathbb{R}^{T_t \times N_v}$. Each input triggers one dense read over all N_v memory positions, with the retrieved content determined by the text queries. The resulting update can propagate through subsequent layers via the residual stream (He et al., 2016; Veit et al., 2016). Depth and frequency ablations are reported in Sec. 3.3.1.

Controlled residual integration. Directly adding Z_{ℓ_r} can destabilize the text residual stream (Alayrac et al., 2022; Gao et al., 2023; Ye et al., 2023). We therefore adapt and bound the update with three controls, each ablated in App. B.2.5.

Channel-wise scaling. Different hidden channels may require different correction strengths. We therefore scale Z_{ℓ_r} elementwise with a learnable vector $\boldsymbol{\alpha}_{\ell_r} \in \mathbb{R}^d$, yielding $\boldsymbol{\alpha}_{\ell_r} \odot Z_{\ell_r}$. The vector is initialized at zero, so the recovery route initially leaves the text stream unchanged.

Norm capping. To prevent the channel-scaled residual from dominating the text representation, we cap its mean token-wise norm relative to that of the text states:

$$\Delta_{\ell_r} = \min\left(1,\ \frac{\kappa\, \bar{n}(H^{\mathrm{text}}_{\ell_r})}{\bar{n}(\boldsymbol{\alpha}_{\ell_r} \odot Z_{\ell_r}) + \delta_0}\right) (\boldsymbol{\alpha}_{\ell_r} \odot Z_{\ell_r}), \qquad \kappa, \delta_0 \in \mathbb{R}_{>0}, \tag{10}$$

where $\bar{n}(X) := \frac{1}{T_t} \sum_{t=1}^{T_t} |X_t|_2$ is the mean token-wise norm, κ bounds the residual-to-text norm ratio, and δ_0 ensures numerical stability.

Entropy-adaptive scaling. To adapt the injection strength across inputs, we use the normalized mean row-wise entropy of A_{ℓ_r} as a heuristic measure of routing uncertainty:

$$e_{\ell_r} = \frac{1}{T_t \log N_v} \sum_{t=1}^{T_t} \mathcal{H}\big(A_{\ell_r}[t,:]\big) \in [0,1], \tag{11}$$

where $\mathcal{H}(\cdot)$ is Shannon entropy, normalized by its maximum $\log N_v$. We assign stronger injection to higher entropy, following the heuristic that diffuse routing may require a larger modality correction:

$$\gamma_{\ell_r} = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\, \sigma(\eta\,(e_{\ell_r} - \tau)), \quad \gamma_{\min}, \gamma_{\max}, \tau \in \mathbb{R}, \quad \eta \in \mathbb{R}_{>0}. \tag{12}$$

where $\gamma_{\min} \le \gamma_{\max}$ bound the injection strength, $\sigma(\cdot)$ is the sigmoid function, η controls transition sharpness, and τ is the entropy threshold. Entropy controls the injection strength, while channel-wise scaling and norm capping constrain the residual. Together, they yield the stabilized update:

$$\tilde{H}^{\mathrm{text}}_{\ell_r} = H^{\mathrm{text}}_{\ell_r} + \gamma_{\ell_r}\, \Delta_{\ell_r}, \qquad \gamma_{\ell_r} \in \mathbb{R}, \quad \Delta_{\ell_r} \in \mathbb{R}^{T_t \times d}. \tag{13}$$

Only the text states are updated; the concatenated state $[\, H^{\mathrm{mod}}_{\ell_r};\ \tilde{H}^{\mathrm{text}}_{\ell_r}\,]$ is then passed to layer ℓ_{r+1}.

Prefill and cached decoding. The equations above describe a full forward pass. At inference, V_m is read once at ℓ_r during prefill. Its effect is then carried through the decoder states and KV cache, with no further recovery reads or memory tokens added to the decoder sequence (App. D).

3 Experiment

We evaluate ENCORE across three modality–language families—vision, proteins, and molecules—to test whether its dual-stage conditioning improves the performance–efficiency trade-off under constrained multimodal token budgets.

3.1 Experimental Setups

Evaluation scope. We evaluate each modality on multiple benchmarks spanning different capabilities and settings. The encoder and decoder training regimes range from frozen to LoRA to full fine-tuning, covering different levels of model plasticity (details in App. A).

Table 1: Results on vision–language benchmarks across three LLM backbones. SQA^I denotes the image-only subset of ScienceQA. POPE and MMMU are reported in percentage points ($\times 100$). The perception-dominant vs. reasoning-dominant grouping follows the dominant-bottleneck criterion detailed in Appendix A.4. Best in **bold**, second-best underlined. ↑ indicates higher is better.

		Perception-dominant					*Reasoning-dominant*			*Overall*
LLM	Method	GQA ↑	MME-P ↑	POPE ↑	TextVQA ↑	SQA^I ↑	MME-C ↑	MMMU ↑	SQA ↑	Norm-Avg ↑
Qwen-2.5	MLP	**58.62**	1267.51	87.25	**48.17**	59.25	231.42	33.4	59.73	82.47
	Q-Former	48.27	982.98	83.99	33.43	55.92	**247.14**	32.3	58.45	74.29
	ENCORE	57.04	**1288.30**	**87.34**	46.65	**59.74**	236.43	**33.6**	**60.43**	**82.52**
StableLM-2	MLP	**60.70**	1319.33	86.84	**56.00**	65.34	215.00	32.6	64.54	85.68
	Q-Former	52.62	1218.86	85.53	45.09	65.69	**255.00**	32.5	64.88	82.69
	ENCORE	59.64	**1338.89**	**87.11**	52.73	**66.04**	245.00	**32.8**	**65.08**	**86.53**
Phi-2	MLP	**61.67**	1451.90	87.99	**57.56**	71.64	281.79	36.6	75.15	94.19
	Q-Former	51.11	1188.83	86.09	43.70	70.10	289.29	35.8	75.38	86.80
	ENCORE	59.66	**1453.32**	**88.21**	55.26	**71.84**	**291.78**	**37.8**	**76.18**	**94.37**

Baselines. Within each modality, all methods share the same modality encoder, LLM backbone, training data, and optimization recipe, differing only in connector design. We use two common connector families as the main references: **MLP**, a two-layer GELU projector that preserves all modality tokens and is widely used in multimodal systems such as TinyLLaVA (Zhou et al., 2024), Prot2Text-V2 (Fei et al., 2025), and DrugChat (Liang et al., 2024); and **Q-Former** (Li et al., 2023a), which compresses encoder outputs into a fixed set of query tokens. These provide shared full-token and compressed references across modalities. We further compare with stronger query-based, token-compression, pruning, and decoder-side reinjection methods in App. B.1.

Evaluation metrics. We report standard metrics for each benchmark (App. A.4). To summarize performance across heterogeneous benchmarks, we additionally report *Norm-Avg*, an anchor-normalized average defined in App. A.5, alongside the original benchmark metrics. For efficiency, we report per-sample FLOPs, throughput, and peak GPU memory.

3.2 Better Accuracy–Efficiency Trade-offs Across Modalities

Table 1–3 and Figure 2 present the full results. ENCORE improves the accuracy–efficiency trade-off across all three modalities, with the largest gains on protein and molecule tasks. The component analyses below examine how compression and recovery contribute to these gains.

3.2.1 Vision–Language

ENCORE achieves the highest overall Norm-Avg on all three vision backbones. Table 1 reports results on six benchmarks—GQA (Hudson & Manning, 2019), MME-P/C (Fu et al., 2025), MMMU (Yue et al., 2024), POPE (Li et al., 2023b), ScienceQA (Lu et al., 2022), and TextVQA (Singh et al., 2019)—under the perception-versus-reasoning categorization from MME (Fu et al., 2025) (criteria in App. A.4). We compare ENCORE (128 visual tokens) with MLP (729 tokens) and Q-Former (128 tokens) on Qwen-2.5-0.5B, StableLM-2-1.6B, and Phi-2-2.7B.

Perception–reasoning trade-off under compression. A consistent pattern is that ENCORE benefits more from reasoning-dominant and robustness-sensitive evaluation, while MLP remains stronger on dense grounding and OCR-heavy tasks. In particular, ENCORE is consistently strongest on MMMU and POPE across all three backbones, whereas MLP retains a clear advantage on GQA and TextVQA. Under the matched 128-token budget, ENCORE also consistently outperforms Q-Former, with the largest Norm-Avg margin on Qwen-2.5 (+8.23), showing that the gain cannot be explained by token count alone. We analyze this perception–reasoning split in Section 3.3.2.

Comparison with stronger baselines. ENCORE remains competitive against a broad set of stronger connector baselines. Among compression-based methods, it outperforms deeper Q-Former, Perceiver Resampler (Alayrac et al., 2022), and mPLUG-Owl (Ye et al., 2024) across all three backbones(Apps. B.1.1 and B.1.2). It also surpasses more recent efficient methods, including Adaptive-pool (Yao et al., 2024), C-Abstractor (Cha et al., 2024), TokenPacker (Li et al., 2025), and FastV/FastV-FT (Chen et al., 2024). On Qwen-2.5-0.5B and 7B, ENCORE exceeds TokenPacker by 1.84 and 1.20 Norm-Avg and 128-token FastV-FT by 4.90 and 4.93 points, while requiring only 4.48

Table 2: Results on the protein-to-text benchmark. All scores are reported in percentage points ($\times 100$). Best in **bold**, second-best underlined. ↑ indicates higher is better.

Method	*Text Generation Metrics*						*Retrieval Metrics*		*Overall*
	E-Match ↑	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	RBT ↑	BBT ↑	Norm-Avg ↑
MLP	**26.17**	26.74	22.99	42.53	33.99	40.38	89.13	81.61	89.58
Q-Former	21.86	**32.20**	27.40	42.24	32.53	39.61	89.01	81.68	90.75
ENCORE	24.39	32.01	**27.94**	**43.37**	**34.06**	**40.97**	**89.36**	**82.27**	**93.34**

Table 3: Results on molecule–language benchmarks. All scores are reported in percentage points ($\times 100$). Best in **bold**, second-best underlined. ↑ indicates higher is better.

Method	*ChemBL*			*Cytotoxicity*			*DrugBank*		*Overall*
	Macro-F1 ↑	AUROC ↑	AUPRC ↑	Macro-F1 ↑	AUROC ↑	AUPRC ↑	Sim. ↑	METEOR ↑	Norm-Avg ↑
MLP	54.88	73.20	34.35	47.64	42.38	11.31	40.46	15.31	80.88
Q-Former	**57.25**	76.86	40.67	47.64	41.83	8.76	41.75	16.11	82.66
ENCORE	54.06	**78.25**	**45.36**	**48.16**	**50.97**	**14.84**	**42.24**	**16.57**	**89.82**

inference TFLOPs at 7B versus 5.99 and 5.83, respectively (App. B.1.3). Finally, against ViCA's multi-layer re-injection (Liu et al., 2026), ENCORE achieves comparable accuracy with a single recovery read while using only 34–61% of its inference FLOPs (App. B.1.2).

3.2.2 Protein–Language

ENCORE leads overall across generation and retrieval. Table 2 reports results using text generation metrics (E-Match, BLEU, ROUGE) and bidirectional retrieval F1 (RBT/BBT). We compare ENCORE (64 tokens) with MLP (18–1023 tokens, $\sim$349 on average) and Q-Former (64 tokens). ENCORE achieves the best overall Norm-Avg (93.34), outperforming Q-Former and MLP by 2.59 and 3.76 points, respectively. It also ranks first on BLEU-4, ROUGE-1/2/L, and both retrieval metrics.

Dense protein-token retention offers limited benefit. Unlike vision, where MLP still benefits dense perceptual tasks, ENCORE ranks first on 6 of 8 metrics. This suggests that retaining all modality tokens provides limited additional benefit in this protein evaluation. Token-usage probes in App. B.4.1 further examine redundancy in the decoder interface, while the recovery-memory probes below test whether complementary information remains available outside the summary.

3.2.3 Molecule–Language

ENCORE shows the largest gains in molecule tasks. Table 3 reports results on ChEMBL, cytotoxicity classification, and DrugBank generation. This setting uses the smallest query budget: ENCORE and Q-Former use only 8 query tokens, while MLP preserves all encoder tokens ($\sim$28 on average; range 1–239). ENCORE achieves a Norm-Avg of 89.82, outperforming Q-Former and MLP by 7.16 and 8.94 points, respectively, and ranks first on 7 of 8 metrics across classification and generation.

Recovery contributes more under stronger compression. At the fixed eight-token molecule interface, adding recovery raises Norm-Avg from 76.26 to 89.82 (+13.56), compared with +1.78 in the 128-token vision setting (App. B.2.3). This larger gain is consistent with our theoretical analysis: stronger compression creates a tighter information bottleneck, increasing the potential value of re-accessing pre-compression modality features. Additional eight-token projector and single-/multi-layer decoder-access controls provide further evidence for this pattern (App. B.1.3).

3.2.4 Efficiency

Figure 2 summarizes the performance–FLOPs trade-off and wall-clock training cost across modalities. In our settings, the efficiency gains are most pronounced for larger decoders and longer modality sequences. Detailed measurements are reported in Appendix B.3.

Lower inference FLOPs than MLP. Figure 2(a)–(c) shows that ENCORE consistently improves the performance–FLOPs trade-off over MLP. On vision with Phi-2, for example, ENCORE achieves a higher Norm-Avg (94.37 vs. 94.19) with less than 40% of the inference FLOPs (2.01 vs. 5.37

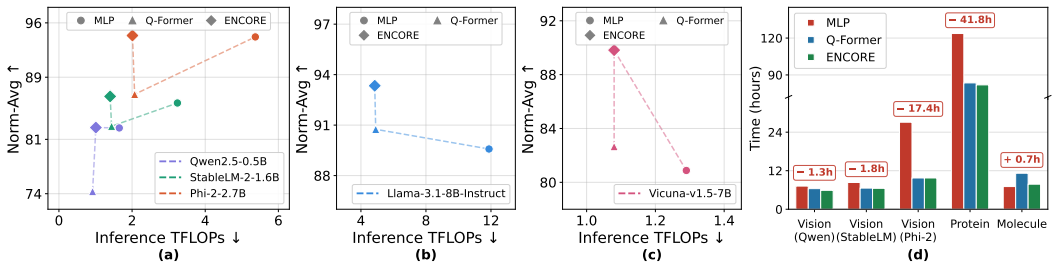

Figure 2: Performance–efficiency trade-off across three modalities and training cost. (a)–(c) Norm-Avg (↑ better) vs. inference TFLOPs (↓ better) for Vision, Protein, and Molecule; the upper-left region indicates better trade-offs. Dashed lines connect the same backbone across connectors. (d) Total wall-clock training time. Red annotations show ENCORE–MLP training time differences.

TFLOPs). The relative savings increase with decoder scale, while ENCORE remains comparable in FLOPs to Q-Former at matched token budgets and consistently achieves higher performance.

Training-time savings across backbones. The reduced computation also translates into lower wall-clock training time. ENCORE is faster than MLP in four of five settings, saving 1.3 h on Vision–Qwen-2.5, 17.4 h on Vision–Phi-2, and 41.8 h on Protein–Llama-3.1-8B. The exception is molecule, where the full-token sequence is already short (∼28 tokens on average), leaving little computation to remove. ENCORE is also at least as fast as Q-Former in all five settings.

FLOPs savings translate to larger wall-clock gains at scale. On Qwen-2.5-0.5B, ENCORE and MLP have similar prefill latency (14.61 vs. 14.51 ms). At 7B, however, ENCORE reduces prefill latency from 34.13 to 20.91 ms and TTFT from 47.22 to 35.24 ms while preserving comparable decode throughput (App. B.3.1). The one-time recovery operation remains small at 0.500–0.568 ms across the profiled settings (Apps. B.3.2 and B.3.3).

3.3 In-depth Analysis of ENCORE

We first validate ENCORE's core design choices on the vision setting (Qwen-2.5-0.5B + SigLIP), then use additional recovery controls to examine cross-modality recovery behavior. All ablations keep other settings fixed; per-benchmark values are reported in App. B.2, Tables 19–21.

3.3.1 Ablations on the Core Design Choices

How do compression and recovery contribute? Compression-only ENCORE reaches 80.74 Overall, compared with 74.29 for Q-Former, showing that the compressed interface itself provides a substantially stronger representation. Adding recovery further raises Overall to 82.52 and improves the reasoning group by 4.30 points (Figure 3(a)). The same recovery architecture improves Q-Former, Adaptive-pool, and TokenPacker, while lowering the full-token MLP result (App. B.2.4). Together with the much larger recovery gain in the eight-token molecule setting (+13.56 Overall) than in the 128-token vision setting (+1.78; App. B.2.3), this pattern is consistent with our theoretical analysis: recovery becomes more valuable as compression creates a tighter information bottleneck. Stabilized integration is also important; removing any of its components lowers Overall (App. B.2.5).

What information does recovery provide? Correct rather than shuffled memory improves held-out sample retrieval from recovery residuals across all three modalities, including for pre-compression memory directions weakly linearly decodable from M_q (App. B.2.6). This indicates that recovery carries sample-specific complementary information beyond the compressed summary. Generic feature injection does not reproduce the same effect: DeepStack-style position-aligned injection with one or four groups reaches 80.24 and 79.29 Overall, respectively, compared with 80.74 for compression alone and 82.52 for ENCORE (App. B.1.4).

How many query tokens are needed? The full-model sweep in Figure 3(b) identifies $N_q = 128$ as the best overall setting, which we adopt as the vision default. Larger budgets do not consistently improve overall performance; their different effects on perception and reasoning are examined below.

Where should recovery occur? Figure 3(c) favors intermediate recovery. We use $\ell_r = \lfloor L/2 \rfloor$ as a shared default rather than tuning depth for each modality or backbone. The molecule sweep shows

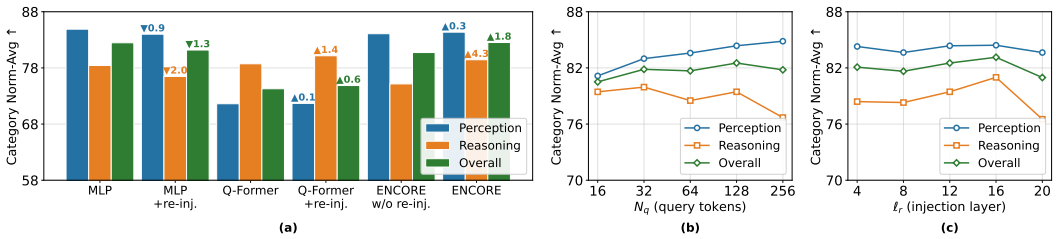

Figure 3: Ablation study on ENCORE design choices, evaluated on Perception, Reasoning, and Overall (Norm-Avg). (a) Path contribution: MLP-, Q-Former-, and ENCORE-based connectors with and without re-injection. Arrows indicate the Norm-Avg change from adding re-injection; upward arrows denote improvement. (b) Effect of query token budget N_q. (c) Effect of injection layer ℓ_r.

the same preference for intermediate layers (App. B.2.7), while multi-layer injection provides no consistent gain over a single recovery read (App. B.2.8).

3.3.2 Component Roles across Tasks and Modalities

Input capacity supports perception. The full-model query-budget sweep shows that increasing N_q steadily improves perception, while reasoning improves slightly before declining at larger budgets (Figure 3(b)). This links input capacity to the preservation of perceptual detail, consistent with the full-token MLP's advantage on grounding and OCR benchmarks such as GQA and TextVQA.

Recovery contributes more to reasoning. Recovery improves ENCORE's reasoning score by 4.3 points (Figure 3(a)). Reasoning varies substantially with injection depth, while perception remains largely stable (Figure 3(c)). Late injection sharply reduces reasoning performance, suggesting that recovered features need downstream processing. These results indicate that recovery complements the input representation mainly through reasoning gains in vision, consistent with ENCORE's strong performance on reasoning and robustness benchmarks such as MMMU and POPE.

Modality-dependent token use. ENCORE's larger gains on protein and molecule tasks show that the vision trade-off varies across modalities. Protein probes reveal sparse, order-insensitive FM-token usage concentrated in early decoding (App. B.4.1), suggesting redundancy in the decoder's use of the full-token interface. In contrast, dense token access benefits visual grounding and OCR.

Robustness under distribution shift. ENCORE retains its advantage on two protein OOD benchmarks, Mol-Instructions (Fang et al., 2024) and UniProtQA (Luo et al., 2024) (App. B.4.2), and on out-of-domain cytotoxicity classification (Table 3). Cytotoxicity shows the largest gains: 9.14 AUROC and 6.08 AUPRC points over Q-Former, supporting OOD generalization.

4 Related Work

Efficient Multimodal Connectors. Query-based connectors compress modality inputs (Li et al., 2023a), while ViCA uses decoder-side cross-attention (Liu et al., 2026) and DeepStack adds visual features at depth (Meng et al., 2024). ENCORE combines a compact input interface with one text-conditioned recovery read and stabilized residual integration. Appendix B.1 compares these designs and their evaluated configurations; Appendix E provides broader context.

5 Conclusion

ENCORE combines compact input compression with a single text-conditioned recovery route at depth. Across vision, protein, and molecule tasks, it improves aggregate performance over MLP and Q-Former without modality-specific hyperparameter tuning. The design reduces inference FLOPs by 16.3–62.6% relative to MLP, shortens training in four of five settings, and accelerates prefill and time to first token on the 7B vision backbone. Stabilized recovery adds substantial gains to compression alone, especially on molecule tasks, demonstrating the value of the second stage alongside the efficient input interface. These results support ENCORE as a connector design that transfers across modalities with improved performance and computational efficiency. **Limitation.** Like most MLLM methods, ENCORE remains constrained by the quality of the underlying modality encoder and LLM.

AI USE STATEMENT

Generative AI was not used to formulate hypotheses or mathematical claims, design the methodology or experiments, implement the methods, or interpret the experimental results. We also used generative AI for grammar checking and language polishing of the manuscript. All AI-assisted content and experimental artifacts were reviewed by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

Our experiments use public benchmark datasets spanning vision, proteins, and molecules and are limited to computational evaluation, without human-subject studies, biological or chemical synthesis, or wet-lab validation. Although scientific multimodal models may have dual-use applications, our experiments focus on standard benchmark tasks. The generated descriptions and predictions should therefore not be treated as experimentally validated biological or chemical findings.

REPRODUCIBILITY STATEMENT

Section 2 specifies the connector, and Appendix D gives its forward procedure. Appendix A describes the datasets, training settings, and evaluation metrics. Appendix B provides component controls, recovery probes, and separate efficiency protocols, while Appendix C states the assumptions and proofs of the theoretical analysis.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm, 2024. URL https://arxiv.org/abs/2312.06742.

Cheng Chen, Yuyu Guo, Pengpeng Zeng, Jingkuan Song, Peng Di, Hang Yu, and Lianli Gao. From one-to-one to many-to-many: Dynamic cross-layer injection for deep vision-language fusion, 2026. URL https://arxiv.org/abs/2601.10710.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024. URL https://arxiv.org/abs/2403.06764.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *ICLR*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=Tlsdsb6l9n.

Xiao Fei, Michail Chatzianastasis, Sarah Almeida Carneiro, Hadi Abdine, Lawrence P. Petalidis, and Michalis Vazirgiannis. Prot2text-v2: Protein function prediction with multimodal contrastive alignment, 2025. URL https://arxiv.org/abs/2505.11194.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model, 2023. URL https://arxiv.org/abs/2304.15010.

Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. How do llms use their depth?, 2026. URL https://arxiv.org/abs/2510.18871.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016. URL https://arxiv.org/abs/1603.05027.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. URL https://arxiv.org/abs/1902.09506.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention, 2021. URL https://arxiv.org/abs/2103.03206.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, and Lizhuang Ma. Efficient multimodal large language models: a survey. *Visual Intelligence*, 3(1), December 2025. ISSN 2731-9008. doi: 10.1007/s44267-025-00099-6. URL http://dx.doi.org/10.1007/s44267-025-00099-6.

Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, Ying Tai, Wankou Yang, Yabiao Wang, and Chengjie Wang. A survey on benchmarks of multimodal large language models, 2024. URL https://arxiv.org/abs/2408.08632.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023a. URL https://arxiv.org/abs/2301.12597.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal LLM. *International Journal of Computer Vision*, 133:6794–6812, 2025. doi: 10.1007/s11263-025-02491-7. URL https://doi.org/10.1007/s11263-025-02491-7.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023b. URL https://arxiv.org/abs/2305.10355.

Youwei Liang, Ruiyi Zhang, Yongce Li, Mingjia Huo, Zinnia Ma, Digvijay Singh, Chengzhan Gao, Hamidreza Rahmani, Satvik Bandi, Li Zhang, Robert Weinreb, Atul Malhotra, Danielle Ann Grotjahn, Linda Awdishu, Trey Ideker, Michael Gilson, and Pengtao Xie. Multi-modal large language model enables all-purpose prediction of drug mechanisms and properties. *bioRxiv*, 2024. URL https://api.semanticscholar.org/CorpusID:273053019.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference, 2025. URL https://arxiv.org/abs/2405.05803.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a. URL https://arxiv.org/abs/2304.08485.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024a. URL https://arxiv.org/abs/2310.03744.

Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *Computers in Biology and Medicine*, 171:108073, March 2024b. ISSN 0010-4825. doi: 10.1016/j.compbiomed.2024.108073. URL http://dx.doi.org/10.1016/j.compbiomed.2024.108073.

Wenjie Liu, Hao Wu, Xin Qiu, Yingqi Fan, Yihan Zhang, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. Vica: Efficient multimodal llms with vision-only cross-attention, 2026. URL https://arxiv.org/abs/2602.07574.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024c. URL https://arxiv.org/abs/2307.06281.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. MolCA: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15623–15638, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.966. URL https://aclanthology.org/2023.emnlp-main.966/.

Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. ProtT3: Protein-to-text generation for text-based protein understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5949–5966, Bangkok, Thailand, August 2024d. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.324. URL https://aclanthology.org/2024.acl-long.324/.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022. URL https://arxiv.org/abs/2209.09513.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Massimo Hong, Yushuai Wu, Mu Qiao, and Zaiqing Nie. Biomedgpt: An open multimodal large language model for biomedicine. *IEEE Journal of Biomedical and Health Informatics*, 2024.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. DeepStack: Deeply stacking visual tokens is surprisingly simple and effective for LMMs, 2024. URL https://arxiv.org/abs/2406.04334.

Tianfan Peng, Yuntao Du, Pengzhou Ji, Shijie Dong, Kailin Jiang, Mingchuan Ma, Yijun Tian, Jinhe Bi, Qian Li, Wei Du, Feng Xiao, and Lizhen Cui. Can visual input be compressed? a visual token compression benchmark for large multimodal models, 2025. URL https://arxiv.org/abs/2511.02650.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models, 2026. URL https://arxiv.org/abs/2403.15388.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. URL https://arxiv.org/abs/1904.08920.

Xinyuan Song, Keyu Wang, PengXiang Li, Lu Yin, and Shiwei Liu. Demystifying the roles of llm layers in retrieval, knowledge, and reasoning, 2026. URL https://arxiv.org/abs/2510.02091.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. Tokencarve: Information-preserving visual token compression in multimodal large language models, 2025. URL https://arxiv.org/abs/2503.10501.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks, 2016. URL https://arxiv.org/abs/1605.06431.

Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. Protchatgpt: Towards understanding proteins with large language models, 2025a. URL https://arxiv.org/abs/2402.09649.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2024. URL https://arxiv.org/abs/2311.03079.

Xin Wang, Zirui Pan, Hong Chen, and Wenwu Zhu. Divico: Disentangled visual token compression for efficient large vision-language model. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(2):1392–1405, 2026a. doi: 10.1109/TCSVT.2025.3567138.

Yujia Wang, Jihong Guan, Wengen Li, Shuigeng Zhou, and Xuhong Wang. Biobridge: Bridging proteins and language for enhanced biological reasoning with llms, 2026b. URL https://arxiv.org/abs/2602.17680.

Zhicong Wang, Zicheng Ma, Ziqiang Cao, Changlong Zhou, Jun Zhang, and Yiqin Gao. Prot2chat: Protein llm with early-fusion of text, sequence and structure, 2025b. URL https://arxiv.org/abs/2502.06846.

Wenkai Xiang, Zhaoping Xiong, Huan Chen, Jiacheng Xiong, Wei Zhang, Zunyun Fu, Mingyue Zheng, Bing Liu, and Qian Shi. Fapm: functional annotation of proteins using multimodal models beyond structural modeling. *Bioinformatics*, 40(12):btae680, 11 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae680. URL https://doi.org/10.1093/bioinformatics/btae680.

Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. Proteingpt: Multimodal llm for protein property prediction and structure understanding, 2025. URL https://arxiv.org/abs/2408.11363.

Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3):1877–1893, March 2025. ISSN 0162-8828. doi: 10.1109/TPAMI.2024.3507000. URL https://doi.org/10.1109/TPAMI.2024.3507000.

Chenyu Yang, Xuan Dong, Xizhou Zhu, Weijie Su, Jiahao Wang, Hao Tian, Zhe Chen, Wenhai Wang, Lewei Lu, and Jifeng Dai. Pvc: Progressive visual token compression for unified image and video processing in large vision-language models, 2024. URL https://arxiv.org/abs/2412.09613.

Senqiao Yang, Junyi Li, Xin Lai, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Visionthink: Smart and efficient vision language model via reinforcement learning, 2025. URL https://arxiv.org/abs/2507.13348.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. DeCo: Decoupling token compression from semantic abstraction in multimodal large language models, 2024. URL https://arxiv.org/abs/2405.20985.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration, 2023. URL https://arxiv.org/abs/2311.04257.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl: Modularization empowers large language models with multimodality, 2024. URL https://arxiv.org/abs/2304.14178.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models, 2025. URL https://arxiv.org/abs/2406.12275.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi 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 Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

Dingkun Zhang, Shuhan Qi, Yulin Wu, Xinyu Xiao, Xuan Wang, and Long Chen. Fast-slow efficient training for multimodal large language models via visual token pruning, 2026. URL https://arxiv.org/abs/2602.03815.

Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models, 2024.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. URL https://arxiv.org/abs/2304.10592.

Xun Zhu, Zheng Zhang, Xi Chen, Yiming Shi, Miao Li, and Ji Wu. Connector-s: A survey of connectors in multi-modal large language models. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 10836–10844. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/1202. URL https://doi.org/10.24963/ijcai.2025/1202. Survey Track.

APPENDIX CONTENTS

A. Detailed Datasets and Experimental Settings

- A.1 Modality-Specific Setup
- A.2 Stage-1 Training Strategies
- A.3 Implementation Details
- A.4 Benchmark Descriptions
- A.5 Norm-Avg Computation

B. More Experiments and Analyses

- B.1 Closely Related Designs and Comparisons
- B.2 Detailed Ablations and Recovery Analysis
- B.3 Detailed Efficiency Analysis
- B.4 Task-Specific and OOD Analyses

C. Theoretical Foundations of Compression and Recovery in ENCORE

- C.1 Unified Notation Reference
- C.2 Computational Complexity Analysis
- C.3 Local Differential View of Compression and Recovery
- C.4 Empirical Evidence for Near-Maximal Per-Token Rank

D. Full Forward-Pass Procedure

E. Expanded Related Work

F. Broader Impact

A DETAILED DATASETS AND EXPERIMENTAL SETTINGS

Unless explicitly noted, we follow the original benchmark papers and official protocol implementations for dataset splits, preprocessing, and evaluation.

A.1 MODALITY-SPECIFIC SETUP

Table 4 summarizes training data, model components, and evaluation settings across modalities.

Table 4: Training data, model components, and evaluation settings across modalities.

Modality	Category	Dataset / Model	Scale
Vision	Stage 1 Training	BLIP-LAION-CC-SBU	∼558K image-text pairs
	Stage 2 Training	LLaVA v1.5 Mix	∼665K instruction samples
	FM Encoder	SigLIP-So400M/14-384	∼400M params
	LLM Decoder	Qwen-2.5-0.5B / StableLM-2-1.6B / Phi-2	0.5B–2.7B params
	Evaluation	Multi-benchmark (GQA / ScienceQA / TextVQA / POPE / MME / MMMU)	6 benchmarks
Molecule	Training	Multi-source (PubChem, ChEMBL, DrugBank)	∼91K QA pairs
	FM Encoder	5-layer GIN (graph only)	∼1.95M params
	LLM Decoder	Vicuna-v1.5	∼7B params
	Evaluation	DrugBank / ChEMBL / Cytotoxicity	835 / 1,372 / 570
Protein	Stage 1 Training	Prot2Text train (contrastive)	∼248K pairs
	Stage 2 Training	Prot2Text train (instruction)	∼248K pairs
	FM Encoder	ESM-2-t33	∼650M params
	LLM Decoder	Llama-3.1-8B-Instruct	∼8B params
	Evaluation	Prot2Text / UniProtQA / Mol-Instructions	4,203 / 3000 / 3000

Evaluation protocols differ across modalities due to the availability of standardized benchmarks. Vision provides a mature and diverse benchmark ecosystem, enabling comprehensive multi-benchmark

evaluation. In contrast, molecule and protein modalities offer more limited and less unified test coverage, so we report both their primary benchmark results and additional evaluations on shifted or external test sets when available.

Vision. We follow the standard two-stage training pipeline (alignment and instruction tuning) used in TinyLLaVA (Zhou et al., 2024), with identical data formats and preprocessing. Images are encoded by SigLIP and projected into the LLM embedding space. The primary setting uses Qwen-2.5-0.5B + SigLIP, with the same recipe applied to other backbones.

Evaluation is conducted on six widely used benchmarks (GQA, ScienceQA, TextVQA, POPE, MME, MMMU), covering both perception-dominant and reasoning-dominant tasks and providing a broad assessment across diverse visual capabilities.

Molecule. We follow the DrugChat data construction and training protocol (Liang et al., 2024), combining ChEMBL, DrugBank, and PubChem into a unified QA-style dataset. ChEMBL and DrugBank are split into training, validation, and test sets, while PubChem is used for training only.

Unlike the original DrugChat architecture, which combines graph and image encoders, we use a graph-only encoder (GIN) for all experiments. This is a deliberate design choice, and empirically yields stronger and more stable performance in our setting.

For evaluation, we follow the DrugChat protocol and report results on DrugBank, ChEMBL, and Cytotoxicity. DrugBank and ChEMBL provide held-out evaluation under the standard split, while Cytotoxicity, which is included in the original DrugChat benchmark suite, exhibits a more shifted distribution and therefore provides a complementary view of generalization.

Protein. We follow Prot2Text-V2 (Fei et al., 2025) with the official train/validation/test splits, using the same two-stage training procedure (contrastive alignment followed by instruction tuning).

Evaluation includes the standard Prot2Text test set as the primary benchmark, together with UniProtQA from BioMedGPT (Luo et al., 2024) and the protein subset of Mol-Instructions (Fang et al., 2024). For UniProtQA and Mol-Instructions, we randomly sample 3,000 examples from their test splits for evaluation, due to their large scale. These additional datasets differ substantially in task format and supervision style from Prot2Text, providing a useful test of protein-modal generalization beyond the primary training setting.

A.2 Stage-1 Training Strategies

We evaluate two stage-1 objectives. Both train only the connector (modality encoder and LLM frozen) and share the same stage-2 autoregressive instruction tuning.

Autoregressive warm-up (Strategy A).

$$\mathcal{L}_{\mathrm{AR}} = -\sum_{k=1}^{K} \log p(y_k \mid y_{<k}, x_m, \mathrm{prompt}).$$

Contrastive alignment (Strategy B). Given paired modality/text embeddings $\{(\mathbf{z}_i^m, \mathbf{z}_i^t)\}_{i=1}^B$:

$$\mathcal{L}_{\mathrm{ctr}} = \frac{1}{2B} \sum_{i=1}^{B} \left[-\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i^m, \mathbf{z}_i^t)/\tau_c)}{\sum_j \exp(\mathrm{sim}(\mathbf{z}_i^m, \mathbf{z}_j^t)/\tau_c)} - \log \frac{\exp(\mathrm{sim}(\mathbf{z}_i^t, \mathbf{z}_i^m)/\tau_c)}{\sum_j \exp(\mathrm{sim}(\mathbf{z}_i^t, \mathbf{z}_j^m)/\tau_c)} \right],$$

where $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity and τ_c is the contrastive temperature. In the Prot2Text setting, the text embedding $\mathbf{z}_i^t$ is extracted from intermediate decoder layer 16 of the frozen LLM, and contrastive supervision is computed on 2 sequence segments per pair (Table 5).

Modality assignment. Vision uses Strategy A in stage 1; protein uses Strategy B. Molecule experiments follow a single-stage protocol and do not use a separate stage-1 strategy.

A.3 Implementation Details

Table 5 provides method-specific training hyperparameters, and Table 6 lists the shared key ENCORE connector parameters used across modalities. The connector uses the shared controls in Table 6,

including $k = 2$, $N_{\min} = 8$, and $\ell_r = \lfloor L/2 \rfloor$. Evaluated query counts are fixed at 128, 64, and 8 for vision, protein, and molecule, respectively; encoder sequence lengths can vary within a modality. The vision budget sweep and recovery-depth studies are reported in Appendix B.2.1. Training schedules follow the modality-specific protocols above. The Q-Former baseline uses 2 layers, an MLP ratio of 4.0, and dropout of 0.1 across all three modalities. The recovery channel scale is initialized at zero. At inference, recovery runs once during prefill; cached decoder states carry its effect into subsequent generation. The encoder memory is not queried at each decode step.

Table 5: Training hyperparameters across modalities. Q-Former and ENCORE share the same N_q within each modality.

Modality		MLP	Q-Former	ENCORE
Vision	N_q	729	128	128
	Stage 1: 1 epoch, batch 256, lr 10^{-3} *Stage 2*: 1 epoch, batch 128, lr 2×10^{-5}			
Protein	N_q	18–1023 (mean $\approx$ 349)	64	64
	Stage 1 (contrastive): batch 256, 12 epochs, lr 2×10^{-4} *Stage 2 (instruction)*: batch 8, 24 epochs, lr 2×10^{-4} *Shared*: contrastive segments 2, output layer 16; encoder LoRA rank 32, alpha 64			
Molecule	N_q	1–239 (mean $\approx$ 28)	8	8
	Shared: lr 2×10^{-5}, batch 32, 10 epochs			

Table 6: Shared key ENCORE connector parameters across modalities.

Hyperparameter	Value
Low-rank query injection rank r_q	64
Token-budget compression exponent k	2
Token-budget floor $N_{\min}$	8
Recovery layer index ℓ_r	$\lfloor L/2 \rfloor$
Recovery cross-attention heads	8
Entropy threshold τ	0.18
Entropy slope η	8.0
Injection bounds $\gamma_{\min}$ / $\gamma_{\max}$	0 / 0.10
Norm-cap ratio κ	0.10
Norm-cap stabilizer δ_0	10^{-8}

A.4 Benchmark Descriptions

Benchmarks for Vision Tasks. We evaluate on six vision benchmark suites, covering eight reported metrics, where higher is better for all metrics. Table 7 summarizes the benchmarks, evaluation metrics, and the dominant capability bottleneck emphasized in our analysis.

Grouping criterion. Following the official perception–cognition split in MME (Fu et al., 2025), and aligned with broader capability-oriented taxonomies in recent MLLM evaluation works (Li et al., 2024; Xu et al., 2025; Liu et al., 2024c), we group the benchmarks in our evaluation suite by their *dominant bottleneck*. We adopt MME's two-sided dichotomy but relabel *cognition* as *reasoning* to more precisely reflect the bottleneck characterising that group. Perception-dominant benchmarks rely more heavily on fine-grained visual evidence extraction, OCR, or object grounding (e.g., GQA, MME-P, POPE, TextVQA, and SQA^{I}), whereas reasoning-dominant benchmarks place greater weight on knowledge integration and multi-step inference (e.g., MME-C, MMMU, and SQA). While GQA contains compositional multi-hop questions, its primary challenge for current models remains accurate visual grounding across complex scenes rather than the reasoning chain itself. Similarly, although SQA^{I} questions may involve scientific reasoning, the distinguishing bottleneck relative to full SQA is whether the model can extract the necessary visual evidence from the image. This grouping is intended as a coarse analytic lens based on the dominant bottleneck, rather than a strict mutually exclusive taxonomy, since several benchmarks involve both perceptual extraction and reasoning.

Table 7: Overview of the vision benchmarks used in our evaluation.

Benchmark	Group	Metric	Dominant bottleneck
GQA (Hudson & Manning, 2019)	Perception	Accuracy	Fine-grained visual grounding and relation recognition
MME-P (Fu et al., 2025)	Perception	Total score	Visual recognition and OCR
POPE (Li et al., 2023b)	Perception	Accuracy	Object grounding fidelity
TextVQA (Singh et al., 2019)	Perception	Accuracy	Scene-text recognition and grounding
SQA^I (Lu et al., 2022)	Perception	Accuracy	Visual evidence extraction from diagrams or images
MME-C (Fu et al., 2025)	Reasoning	Total score	Commonsense reasoning and numerical calculation
MMMU (Yue et al., 2024)	Reasoning	Accuracy	Knowledge-intensive multi-step inference
SQA (Lu et al., 2022)	Reasoning	Accuracy	Scientific knowledge integration and multi-step reasoning

Evaluation Metrics for Molecule and Protein Tasks We report BLEU-2/4, ROUGE-1/2/L, METEOR, and semantic similarity (cosine similarity between `all-MiniLM-L6-v2` sentence embeddings). Normalized Exact Match (EM) counts matches after lowercasing and stripping punctuation.

Binary classification (molecule). We report Macro-F1 (primary), AUROC, AUPRC (Average Precision), and Accuracy. For yes/no decisions we use probability-based scoring:

$$\text{score} = p_{\text{yes}} - p_{\text{no}}, \qquad \hat{y} = \mathbf{1}[p_{\text{yes}} \geq p_{\text{no}}].$$

Retrieval alignment (protein). Protein-text alignment is evaluated bidirectionally. For each direction we compute precision, recall, and F1 at a fixed retrieval cutoff, then report the bidirectional average:

$$\text{Bi-F1} = \tfrac{1}{2}\left(\text{F1}^{p\to t} + \text{F1}^{t\to p}\right).$$

In result tables, RBT and BBT denote these bidirectional F1 scores.

A.5 Norm-Avg Computation

To compare multi-benchmark performance on a unified scale, we compute anchor-normalized averages. For modality m with benchmarks $\mathcal{B}_m$ (of size B_m), score $s_{m,i}$, and fixed anchor $a_{m,i} > 0$, we define

$$\text{Norm-Avg}_m = \frac{100}{B_m} \sum_{i=1}^{B_m} \frac{s_{m,i}}{a_{m,i}}.$$

Anchors are rounded practical ceilings chosen prior to method comparison and kept fixed across all tables. This definition is applied separately to the vision, molecule, and protein modalities, using the benchmark-specific anchors listed in Table 8.

Table 8: Fixed anchors for Norm-Avg computation.

Vision		Molecule		Protein	
Metric	a_i	Metric	a_i	Metric	a_i
GQA	70	ChEMBL Macro-F1	60	E-Match	30
MME-P	1500	ChEMBL AUROC	80	BLEU-2	35
POPE	90	ChEMBL AUPRC	50	BLEU-4	30
TextVQA	60	Cytotox. Macro-F1	50	ROUGE-1	45
SQA^I	75	Cytotox. AUROC	55	ROUGE-2	35
MME-C	300	Cytotox. AUPRC	20	ROUGE-L	45
MMMU	40	Similarity	45	RBT	90
SQA	80	METEOR	20	BBT	85

A.5.1 Anchor Sensitivity

We test whether the aggregate comparison depends on the choice of normalization anchors. The compared models are MLP, Q-Former, and ENCORE, plus ViCA for vision and molecules. Vision uses Qwen-2.5-0.5B. We consider the fixed anchors in Table 8, observed per-metric maxima, these maxima multiplied by 1.10 or 1.25, and rounded practical upper bounds.

Table 9: Norm-Avg under alternative anchors. "Margin" is the reported ENCORE minus runner-up score; values and margins are rounded separately. The $+10\%$ and $+25\%$ columns rescale all observed-max anchors equally and therefore preserve ordering by construction.

Method	Fixed	Obs. max.	$+10\%$	$+25\%$	Rounded
Vision					
MLP	82.47	97.56	88.69	78.05	93.85
ViCA	82.45	97.58	88.71	78.07	93.81
Q-Former	74.29	87.94	79.94	70.35	84.66
ENCORE	82.52	97.64	88.76	78.11	93.93
Margin	+0.05	+0.06	+0.05	+0.04	+0.08
Molecule					
MLP	80.88	88.95	80.86	71.16	83.23
ViCA	81.38	89.30	81.19	71.44	83.41
Q-Former	82.66	90.50	82.27	72.40	84.49
ENCORE	89.82	99.30	90.28	79.44	92.91
Margin	+7.16	+8.80	+8.01	+7.04	+8.42
Protein					
MLP	89.58	95.09	86.44	76.07	89.58
Q-Former	90.75	96.26	87.51	77.01	90.75
ENCORE	93.34	99.08	90.07	79.26	93.34
Margin	+2.59	+2.82	+2.56	+2.25	+2.59

The margins remain small in vision (0.04–0.08) and larger in molecules (7.04–8.80) and proteins (2.25–2.82). ENCORE remains first under all five anchor schemes, with comparable aggregate vision accuracy and larger gains on the scientific tasks.

We also perturb anchors independently within relative ranges of $\pm1\%$, $\pm3\%$, and $\pm5\%$, with 200,000 Monte Carlo draws per range. Table 10 reports the fraction in which ENCORE ranks first in the evaluated comparison set.

Table 10: ENCORE rank-1 frequency under independent anchor perturbations.

Modality	$\pm1\%$	$\pm3\%$	$\pm5\%$
Vision	1.000	1.000	0.963
Molecule	1.000	1.000	1.000
Protein	1.000	1.000	1.000

Only the closest vision ordering, ENCORE versus ViCA, changes in the reported perturbation analysis. This leaves their arithmetic-cost comparison unchanged.

B More Experiments and Analyses

The following comparisons examine related connector designs, component contributions, practical efficiency, and task-specific behavior.

B.1 Closely Related Designs and Comparisons

Design comparison. Table 11 compares input representations, decoder-side memory access, and writeback locations. ENCORE combines a persistent compact interface with a single text-conditioned update. The table distinguishes published designs from the configurations evaluated in this study.

Evaluation scope. MLP and Q-Former provide shared full-token and compressed references across modalities. Query-based connectors are also used in prior protein and molecule systems (Liu et al., 2023b; 2024d; Xiang et al., 2024). We first compare alternative query-based and decoder-access

Table 11: Vision connector designs and evaluated controls. Input token counts and result references describe the evaluated configurations; published mechanisms are listed separately.

Design	Decoder input	Deep memory source	Read and write locations	Access schedule / evidence
Q-Former / Resampler	128 summary tokens	None	Query-based compression at input	Input stage; Table 12
TokenPacker	144 packed tokens	None	Coarse-to-fine aggregation at input	Input stage; Table 15
ViCA (published) (Liu et al., 2026)	Text	Projected visual embeddings	Visual tokens bypass self-attention and FFNs; text queries access memory	Selected decoder layers
ViCA-based control (evaluated)	729 visual tokens	Pre-compression visual features	Cross-attention updates text positions	Eight layers, shared parameters; Tables 14 and 13
DeepStack (published) (Meng et al., 2024)	Base visual tokens	Additional visual feature groups	Position-aligned addition at visual positions	Successive decoder layers
DeepStack-style controls (evaluated)	729 or 128 tokens	Aligned visual feature groups	Addition at modality positions	One or four groups after compression; Table 18
ENCORE	128 summary tokens	Pre-compression visual memory	Text-conditioned read; stabilized update at text positions	One midpoint read during prefill; Tables 15 and 22

designs in vision, then examine projector compression, pruning, and selected molecule adaptations in Appendix B.1.3.

B.1.1 Additional Single-Connector Baselines

To disentangle the effect of connector *architecture* from connector *capacity*, we extend the comparison in Table 12 with two additional axes of variation.

Setup. Q-Former-4L doubles the depth of the default 2-layer Q-Former to 4 cross-attention layers while keeping the query count fixed at 128. **Resampler** replaces the Q-Former with a Perceiver Resampler (Alayrac et al., 2022) and is tested at 2 and 4 layers (denoted **Resampler-2L** and **Resampler-4L**). All other training details (data, schedule, LLM backbone, visual encoder) are identical to the main experiments.

Results. Table 12 shows that increasing connector depth has mixed effects. Resampler-4L improves over Resampler-2L on Qwen-2.5 but regresses on StableLM-2 and Phi-2; Q-Former-4L also does not consistently improve Overall. ENCORE achieves the highest Overall in these comparisons. Its margin over the strongest single-connector variant is 0.14, 1.47, and 5.94 points across the three backbones.

B.1.2 Comparison with Representative Stronger Baselines

We compare mPLUG-Owl (Ye et al., 2024), which concentrates capacity in input-stage compression, with ViCA (Liu et al., 2026), which adds repeated decoder-side visual access.

Setup. ViCA-based control. Our evaluated configuration retains a 2-layer MLP front-end (identical to the MLP baseline) and injects raw vision patch features into 8 LLM decoder layers via a single *shared* T2V cross-attention block. This control uses a full-token input and additional cross-attention. The injection layers are $\{0, 1, 5, 6, 7, 8, 9, 11\}$ for Qwen-2.5 and StableLM-2, and $\{0, 1, 7, 8, 9, 10, 11, 14\}$ for Phi-2. **mPLUG-Owl** (Ye et al., 2024) replaces Q-Former with a deeper visual abstractor: 6 layers, hidden size 1024, and 128 learnable query tokens, producing a fixed-length 128-token visual sequence that is fed into the LLM with no internal injection. All other training details

Table 12: Results on vision–language benchmarks across three LLM backbones. SQA^I denotes the image-only subset of ScienceQA. POPE and MMMU are reported in percentage points ($\times 100$). The perception-dominant vs. reasoning-dominant grouping follows the dominant-bottleneck criterion detailed in Appendix A.4. Best in **bold**, second-best underlined.

		Perception-dominant					*Reasoning-dominant*			*Overall*
LLM	Method	GQA ↑	MME-P ↑	POPE ↑	TextVQA ↑	SQA^I ↑	MME-C ↑	MMMU ↑	SQA ↑	Norm-Avg ↑
Qwen-2.5	MLP	**58.62**	1267.51	87.25	**48.17**	59.25	231.42	33.4	59.73	82.47
	Q-Former-2L	48.27	982.98	83.99	33.43	55.92	247.14	32.3	58.45	74.29
	Q-Former-4L	49.71	955.82	73.71	33.61	57.11	**273.21**	30.2	57.98	73.48
	Resampler-2L	56.01	1228.82	86.67	46.05	59.59	242.14	33.2	60.36	81.70
	Resampler-4L	56.45	1254.26	86.67	46.61	59.69	233.57	**35.2**	60.27	82.38
	ENCORE	57.04	**1288.30**	**87.34**	46.65	**59.74**	236.43	33.6	**60.43**	**82.52**
StableLM-2	MLP	**60.70**	1319.33	86.84	**56.00**	65.34	215.00	32.6	64.54	85.68
	Q-Former-2L	52.62	1218.86	85.53	45.09	65.69	255.00	32.5	64.88	82.69
	Q-Former-4L	52.26	1223.42	85.95	43.58	65.34	**279.29**	32.3	65.06	83.33
	Resampler-2L	54.31	1299.27	84.91	48.98	65.69	265.36	**33.2**	65.02	85.06
	Resampler-4L	55.15	1291.21	84.31	50.07	65.20	243.93	**33.2**	63.88	84.14
	ENCORE	59.64	**1338.89**	**87.11**	52.73	**66.04**	245.00	32.8	**65.08**	**86.53**
Phi-2	MLP	**61.67**	1451.90	87.99	**57.56**	71.64	281.79	36.6	75.15	94.19
	Q-Former-2L	51.11	1188.83	86.09	43.70	70.10	289.29	35.8	75.38	86.80
	Q-Former-4L	51.03	1140.60	85.08	42.66	69.56	288.93	**38.2**	74.56	86.54
	Resampler-2L	52.58	1243.39	85.13	45.43	**72.14**	280.36	37.8	75.95	88.43
	Resampler-4L	51.68	1178.31	84.91	45.04	72.04	263.57	37.9	75.60	86.87
	ENCORE	59.66	**1453.32**	**88.21**	55.26	71.84	**291.78**	37.8	**76.18**	**94.37**

(data, schedule, LLM backbone, visual encoder) are identical to the main experiments. Performance and efficiency results are reported in Tables 13 and 14.

Table 13: Comparison with stronger reinjection / heavy-compression baselines on vision–language benchmarks across three LLM backbones. ViCA injects raw vision features into 8 internal LLM decoder layers; mPLUG-Owl replaces the connector with a 6-layer, 1024-hidden, 128-query visual abstractor. SQA^I denotes the image-only subset of ScienceQA. POPE and MMMU are reported in percentage points ($\times 100$). Best in **bold**, second-best underlined.

		Perception-dominant					*Reasoning-dominant*			*Overall*
LLM	Method	GQA ↑	MME-P ↑	POPE ↑	TextVQA ↑	SQA^I ↑	MME-C ↑	MMMU ↑	SQA ↑	Norm-Avg ↑
Qwen-2.5	MLP	58.62	1267.51	87.25	48.17	59.25	231.42	33.4	59.73	82.47
	ViCA	**59.74**	1234.55	86.25	**50.13**	**61.43**	217.14	33.0	**60.65**	82.45
	mPLUG-Owl	52.78	1211.95	86.57	44.28	58.65	**236.79**	33.3	59.68	80.15
	ENCORE	57.04	**1288.30**	**87.34**	46.65	59.74	236.43	**33.6**	60.43	**82.52**
StableLM-2	MLP	60.70	1319.33	86.84	**56.00**	65.34	215.00	32.6	64.54	85.68
	ViCA	**61.86**	1326.99	**87.16**	54.59	65.05	238.57	32.6	64.30	**86.60**
	mPLUG-Owl	52.85	1176.03	86.35	45.54	65.44	**247.14**	32.4	64.84	82.18
	ENCORE	59.64	**1338.89**	87.11	52.73	**66.04**	245.00	**32.8**	**65.08**	86.53
Phi-2	MLP	61.67	1451.90	87.99	**57.56**	71.64	281.79	36.6	75.15	94.19
	ViCA	**62.12**	1428.05	**88.24**	57.36	71.34	290.29	36.2	75.90	94.36
	mPLUG-Owl	56.00	1333.96	87.30	51.13	**72.29**	288.21	**38.1**	76.14	91.75
	ENCORE	59.66	**1453.32**	88.21	55.26	71.84	**291.78**	37.8	**76.18**	**94.37**

Results. ENCORE and ViCA are within 0.07 Norm-Avg across the three backbones (Table 13). ViCA is stronger on GQA, whereas ENCORE performs better on several reasoning-group metrics. ENCORE uses 34–61% of ViCA's inference FLOPs in these profiles. This difference reflects both the shorter persistent input sequence and the fewer memory reads.

mPLUG-Owl is 2.3–4.4 points below ENCORE in Overall and uses approximately 109–111M connector parameters, compared with ENCORE's 26–51M (Table 14). Throughput and peak memory vary by setting, so aggregate accuracy and FLOPs are interpreted alongside those practical measurements.

Table 14: Efficiency comparison with stronger SOTA connector baselines on vision benchmarks. #Inj denotes the number of vision-feature injections into LLM decoder layers; the in-connector designs of each baseline are described in Appendix B.1.2. Conn. (M) denotes the total number of trainable parameters introduced by the connector module or connector-side adaptation for each method, in millions. Per-sample TFLOPs (10^{12}), throughput (samples/s), and peak GPU memory (GiB) are reported under a unified protocol with batch size 1.

				Training			Inference		
LLM Backbone	Method	#Inj	Conn. (M)	TFLOPs↓	Thru.↑	Mem.↓	TFLOPs↓	Thru.↑	Mem.↓
Qwen-2.5-0.5B	MLP	0	1.84	5.32	5.72	31.21	1.65	2.67	11.98
	ViCA	8	6.09	5.39	4.39	31.33	1.65	2.05	12.03
	mPLUG-Owl	0	108.97	3.46	3.81	31.28	1.06	2.04	12.01
	ENCORE	1	25.79	3.34	5.60	29.37	1.01	2.65	11.79
StableLM-2-1.6B	MLP	0	6.56	10.43	4.56	13.86	3.24	2.13	5.33
	ViCA	8	25.71	11.42	4.93	13.96	3.50	2.30	5.37
	mPLUG-Owl	0	110.15	5.48	4.64	13.60	1.68	2.48	5.23
	ENCORE	1	40.77	4.63	4.64	12.70	1.40	2.20	5.10
Phi-2-2.7B	MLP	0	9.51	17.29	3.32	20.72	5.37	1.55	7.97
	ViCA	8	38.70	19.09	3.60	20.88	5.85	1.65	8.03
	mPLUG-Owl	0	110.67	8.13	3.42	21.21	2.49	1.88	8.16
	ENCORE	1	50.83	6.65	3.48	20.04	2.01	1.68	8.05

B.1.3 Projector Compression, Pruning, and Backbone Scaling

We compare ENCORE with Adaptive-pool following DeCo (Yao et al., 2024), C-Abstractor from Honeybee (Cha et al., 2024), and TokenPacker (Li et al., 2025). We also evaluate FastV (Chen et al., 2024) applied after training, its pruning-aware adaptation (FastV-FT), and ViCA. Within each modality and backbone, these comparisons share the encoder, training data, and evaluation recipe. FastV-FT uses the same pruning rule and retained-token budget as FastV; its reported training FLOPs retain the full-token cost.

Extended comparison protocol. ENCORE uses $N_q = 128$ visual tokens. The grid-based vision projectors output 144 tokens from the 27×27 encoder grid. FastV token counts denote retained tokens after pruning, so its early layers still process the longer sequence.

Training and inference FLOPs in the following tables use the H200 profiling series. "TF" denotes training-free pruning of an existing trained MLP checkpoint. Table 15 combines the projector and pruning comparisons at two vision backbone scales.

TokenPacker is the strongest added projector at both scales. ENCORE exceeds it by 1.84 Overall at 0.5B and 1.20 at 7B, while MLP and ViCA remain close to ENCORE in aggregate accuracy. Pruning-aware adaptation improves over aggressive post-hoc pruning at both scales but does not close that gap. At 7B, ENCORE uses 4.48 inference TFLOPs versus 13.53 for MLP and ViCA.

Molecule transfer. We also evaluate the projector variants with eight molecule tokens (Table 16) and transfer the pruning and ViCA controls (Table 17). Classification and generation denote the normalized metric groups.

At eight molecule tokens, ENCORE leads the tested compressed projectors and exceeds eight-layer ViCA by 8.44 Overall. Its FLOPs are 29.1% lower than MLP's in this profiling series. Removing recovery changes performance substantially while leaving the two-decimal FLOPs unchanged; the separately measured route cost is nonzero (Appendix B.3.3).

B.1.4 Decoder-Side Access after Compression

We compare query-conditioned recovery with DeepStack-style position-aligned residual addition (Meng et al., 2024). The vision controls use the same Qwen-2.5-0.5B stack. The full-token version injects visual features into visual-token positions; the compressed variants apply this form of injection with one or four feature groups.

Table 15: Extended vision comparisons. Perception, reasoning, and Overall are Norm-Avg; Train and Infer are TFLOPs. Token counts follow §B.1.3.

Method	Tokens	Perc.	Reas.	Overall	Train	Infer
Qwen-2.5-0.5B						
MLP	729	84.89	78.43	82.47	5.32	1.65
ViCA	729	85.79	76.90	82.45	5.39	1.65
Q-Former	128	71.62	78.73	74.29	3.06	0.92
Adaptive-pool	144	80.65	78.44	79.82	3.35	1.02
C-Abstractor	144	72.01	77.79	74.18	3.36	1.02
TokenPacker	144	82.82	77.12	80.68	3.38	1.02
FastV (364)	364	82.21	79.14	81.06	TF	1.28
FastV (128)	128	75.36	76.71	75.86	TF	1.04
FastV-FT (128)	128	78.11	76.81	77.62	5.32	1.04
ENCORE	128	84.36	79.45	82.52	3.34	1.01
Qwen-2.5-7B						
MLP	729	95.30	102.43	97.97	43.58	13.53
ViCA	729	95.02	103.36	98.15	44.17	13.53
Q-Former	128	82.39	102.69	90.00	17.55	5.27
Adaptive-pool	144	96.55	95.64	96.21	18.12	5.99
C-Abstractor	144	89.30	92.85	90.63	18.13	5.99
TokenPacker	144	97.35	96.31	96.96	18.15	5.99
FastV (364)	364	91.57	105.47	96.78	TF	8.84
FastV (128)	128	83.77	102.56	90.82	TF	5.83
FastV-FT (128)	128	84.12	108.42	93.23	43.58	5.83
ENCORE	128	94.08	104.97	98.16	14.80	4.48

Table 16: Eight-token projector comparisons on molecules. Group scores and Overall use Norm-Avg.

Method	Tokens	Class.	Gen.	Overall
MLP	~ 28	80.09	83.23	80.88
Q-Former	8	81.33	86.66	82.66
Adaptive-pool	8	72.44	80.00	74.33
C-Abstractor	8	75.71	82.64	77.44
TokenPacker	8	72.15	81.95	74.60
ENCORE	8	90.30	88.36	89.82

Table 17: Molecule pruning, recovery, and decoder-access controls under the H200 FLOPs series. Train and Infer are TFLOPs; TF denotes post-hoc use of a trained full-token checkpoint.

Method	Tokens	Overall	Train	Infer
MLP	~ 28	80.88	3.51	1.17
ViCA (8 layers)	~ 28	81.38	3.51	1.17
ViCA (1 layer)	~ 28	78.57	3.07	1.04
FastV (half)	~ 14	80.74	TF	1.00
FastV (8)	8	75.15	TF	0.93
FastV-FT (8)	8	77.20	3.51	0.93
ENCORE w/o recovery	8	76.26	2.49	0.83
ENCORE	8	89.82	2.49	0.83

Table 18: Decoder-access controls on full-token and compressed interfaces. Infer is TFLOPs and prefill is milliseconds; a dash denotes an unreported timing.

Setting	Tokens	Overall	Infer	Prefill
MLP	729	82.47	1.65	14.51
DeepStack-style, full	729	82.94	1.66	16.36
Compression only	128	80.74	1.01	—
Compressed, 1 group	128	80.24	1.01	13.83
Compressed, 4 groups	128	79.29	1.01	15.00
ENCORE	128	82.52	1.01	14.61

Table 19: Path-contribution ablation on vision benchmarks (Qwen-2.5-0.5B + SigLIP). MLP uses 729 visual tokens, while compression-based settings use 128 query tokens. All reported scores are in percentage points ($\times 100$). Best in **bold**, second-best underlined. Perception/Reasoning grouping follows Appendix A.4; Overall is Norm-Avg (Appendix A.5).

	Perception-dominant					*Reasoning-dominant*			Norm-Avg		
Method	**GQA↑**	**MME-P↑**	**POPE↑**	**TextVQA↑**	**SQA^{I}↑**	**MME-C↑**	**MMMU↑**	**SQA↑**	**Perc.↑**	**Reas.↑**	**Overall↑**
MLP	**58.62**	1267.51	87.25	**48.17**	59.25	231.42	33.4	59.73	**84.89**	78.43	82.47
MLP + re-inj.	56.96	1269.57	86.54	47.43	59.10	222.86	32.2	59.73	84.00	76.48	81.18
Q-Former	48.27	982.98	83.99	33.43	55.92	247.14	32.3	58.45	71.62	78.73	74.29
Q-Former + re-inj.	48.41	981.40	83.69	33.75	55.94	**247.93**	33.3	59.67	71.68	**80.16**	74.86
ENCORE w/o re-inj.	56.28	**1305.19**	87.33	45.88	59.64	210.00	32.0	60.36	84.09	75.15	80.74
ENCORE	57.04	1288.30	**87.34**	46.65	**59.74**	236.43	**33.6**	**60.43**	84.36	79.45	**82.52**

DeepStack-style injection improves the full-token control. After compression, its one- and four-group variants score below compression alone, whereas ENCORE improves Overall by 1.78 points with text-conditioned recovery and stabilized integration.

B.2 Detailed Ablations and Recovery Analysis

The core vision tables corresponding to Figure 3 are collected first in Appendix B.2.1. Component attribution is summarized in Appendix B.2.2; cross-modality and cross-interface controls follow in Appendices B.2.3 and B.2.4. Appendices B.2.5 and B.2.6 report stabilization and memory probes. Molecule depth, recovery frequency, and local inference sensitivity are reported in Appendices B.2.7, B.2.8, and B.2.9.

B.2.1 Core Vision Ablations

Tables 19–21 provide the per-benchmark results for Figure 3(a)–(c): path contribution, query-token budget, and recovery depth. All three studies use Qwen-2.5-0.5B with SigLIP and the same training data and optimization recipe.

Path contribution. Table 19 compares MLP, Q-Former, and ENCORE with and without recovery. It reports both the aggregate gains and their perception/reasoning breakdown, including ENCORE's 80.74 to 82.52 Overall improvement and 4.30-point reasoning gain.

Query-token budget. Table 20 varies $N_q \in \{16, 32, 64, 128, 256\}$ at fixed recovery depth, corresponding to Figure 3(b). The 128-token setting has the highest Overall score and is the vision default.

Recovery depth. Table 21 varies $\ell_r \in \{4, 8, 12, 16, 20\}$ at $N_q = 128$, corresponding to Figure 3(c). The midpoint provides the shared depth rule used across modalities and backbones.

B.2.2 Compression and the Complete Recovery Route

Table 22 brings together the compression-only, recovery, and integration configurations on Qwen-2.5-0.5B with SigLIP. Compression-only ENCORE and Q-Former both use $N_q = 128$. The fixed-strength configuration uses $\gamma = 0.10$; the component-removal settings are described in Appendix B.2.5.

Table 20: Ablation on query-token budget N_q with fixed route injection layer $\ell_r = 12$ (Qwen-2.5-0.5B + SigLIP). All reported scores are in percentage points ($\times 100$). Best in **bold**, second-best underlined. Perception/Reasoning grouping follows Appendix A.4; Overall is Norm-Avg (Appendix A.5).

	Perception-dominant					*Reasoning-dominant*			Norm-Avg		
Setting	**GQA↑**	**MME-P↑**	**POPE↑**	**TextVQA↑**	**SQA^{I}↑**	**MME-C↑**	**MMMU↑**	**SQA↑**	**Perc.↑**	**Reas.↑**	**Overall↑**
$N_q = 16$	54.35	1214.57	85.71	43.81	59.15	**250.00**	32.1	59.80	81.15	79.44	80.51
$N_q = 32$	55.68	1258.29	85.77	45.98	59.69	229.29	**35.1**	**60.53**	82.99	**79.95**	81.85
$N_q = 64$	56.56	1244.40	86.55	46.70	**60.14**	232.86	32.9	**60.53**	83.59	78.51	81.68
$N_q = 128$	57.04	**1288.30**	**87.34**	46.65	59.74	236.43	33.6	60.43	84.36	79.45	**82.52**
$N_q = 256$	**57.54**	1287.13	87.19	**47.87**	59.69	223.57	32.1	60.25	**84.85**	76.70	81.79

Table 21: Ablation on route injection layer index ℓ_r with fixed token budget $N_q = 128$ (Qwen-2.5-0.5B + SigLIP). All reported scores are in percentage points ($\times 100$). Best in **bold**, second-best underlined. Perception/Reasoning grouping follows Appendix A.4; Overall is Norm-Avg (Appendix A.5).

	Perception-dominant					*Reasoning-dominant*			Norm-Avg		
Setting	**GQA↑**	**MME-P↑**	**POPE↑**	**TextVQA↑**	**SQA^{I}↑**	**MME-C↑**	**MMMU↑**	**SQA↑**	**Perc.↑**	**Reas.↑**	**Overall↑**
$\ell_r = 4$	56.78	1255.85	87.44	47.24	**60.54**	231.79	32.9	60.55	84.29	78.40	82.08
$\ell_r = 8$	56.40	1252.46	86.70	46.50	60.24	227.14	33.4	60.55	83.64	78.30	81.64
$\ell_r = 12$	57.04	**1288.30**	87.34	46.65	59.74	236.43	33.6	60.43	84.36	79.45	82.52
$\ell_r = 16$	56.87	1257.41	**87.77**	**47.44**	60.34	**247.50**	**33.9**	**60.58**	**84.42**	**80.99**	**83.14**
$\ell_r = 20$	**57.20**	1240.77	87.24	46.82	59.10	221.43	32.4	59.80	83.64	76.52	80.97

Compression-only ENCORE gains 6.45 Overall over Q-Former but loses 3.58 on reasoning. Recovery then adds 1.78 Overall and 4.30 on reasoning. The complete route also exceeds the fixed-strength configuration (80.28). Table 19 gives the full per-benchmark path ablation, including recovery added to MLP and Q-Former.

B.2.3 Recovery across Bottleneck Settings

We compare ENCORE with and without recovery while holding the encoder, backbone, data, training protocol, and token budget fixed within each modality. Table 23 contrasts the 128-token vision interface with the eight-token molecule interface.

Recovery contributes more in the eight-token molecule setting than in the 128-token vision setting. The query-budget sweep in Table 20 separately examines the effect of token count on complete ENCORE models.

B.2.4 Recovery across Compressed Interfaces

We attach the same recovery architecture to different input interfaces in the Qwen-2.5-0.5B vision setting. Each pair shares its encoder, training recipe, and evaluation protocol. Table 24 reports the resulting change in Overall.

Table 22: Compression, recovery, and stabilization controls. Perception, reasoning, and Overall are Norm-Avg scores.

Setting	Perc.	Reas.	Overall
Q-Former	71.62	78.73	74.29
ENCORE w/o recovery	84.09	75.15	80.74
Fixed strength	83.72	74.54	80.28
w/o entropy scaling	84.06	75.82	80.97
w/o norm cap	84.15	77.82	81.78
w/o channel gate	84.11	78.63	82.06
Complete ENCORE	84.36	79.45	82.52
Full-token MLP	84.89	78.43	82.47

Table 23: Matched within-modality recovery controls. Scores are Overall Norm-Avg.

Modality	N_q	w/o recovery	w/ recovery	Gain
Vision	128	80.74	82.52	+1.78
Molecule	8	76.26	89.82	+13.56

Table 24: Recovery controls across interfaces. ENCORE uses $N_q = 128$ visual tokens.

Interface	Tokens	w/o	w/	Gain
Full-token MLP	729	82.47	81.18	-1.29
Q-Former	128	74.29	74.86	+0.57
Adaptive-pool	144	79.82	81.04	+1.22
TokenPacker	144	80.68	81.90	+1.22
ENCORE compression	128	80.74	82.52	+1.78

Recovery improves all four tested compressed interfaces but lowers the full-token result. This pattern supports the role of recovery in supplying complementary information to compressed interfaces.

B.2.5 Route Stabilization Controls

Setup. The recovery route combines three control mechanisms: entropy-adaptive scaling (γ, Eq. (12)), norm capping (κ, Eq. (10)), and channel-wise gating. We train configurations with each control removed in turn on the vision setting (Qwen-2.5-0.5B + SigLIP, $N_q = 128$, $\ell_r = 12$): *w/o entropy* fixes $\gamma = \gamma_{\max}$; *w/o normcap* sets $\kappa \to \infty$; and *w/o gate* fixes the channel-wise gate to all ones. Results are reported in Table 25.

Analysis. All three components improve the overall route, with the effect concentrated on reasoning. Perception Norm-Avg varies by only 0.30 points across ablations, whereas Reasoning Norm-Avg drops by up to 3.63 points when entropy-adaptive scaling is removed. Removing entropy-adaptive scaling produces the largest Overall drop (-1.55 Overall, -3.63 Reas.), with the largest single-benchmark drop on MME-C (-20.0). Norm capping ranks second (-0.74 Overall, -1.63 Reas.) and causes the largest drop on MMMU (-1.8), suggesting that magnitude control is especially important for knowledge-intensive reasoning. Channel-wise gating has the smallest overall effect (-0.46 Overall), but still improves the final overall and reasoning balance. Although some ablated variants slightly improve individual perception metrics, none matches the full stabilized route in Overall or Reasoning Norm-Avg.

B.2.6 Sample-Specific Recovery-Memory Probes

We test whether the recovered residual carries the current sample's modality information. A linear probe predicts training-fold PCA coordinates of the memory supplied to the recovery route, using held-out evaluation and three shuffle controls. The vision target is pre-compression visual memory; the protein and molecule targets are query-token representations. Correct memory uses the current sample; shuffled memory changes only the source memory. We report Top-1/Top-5 retrieval and mean reciprocal rank (MRR).

A second probe retains four memory directions with weak linear decodability. Vision directions are selected by decodability from the input summary; protein and molecule directions are selected by decodability from the decoder state immediately before injection, using cross-fitting. Direction selection uses training folds only. Table 27 reports retrieval on these directions and one-sided exact McNemar tests comparing paired Top-1 outcomes. "C-only/S-only" counts cases correct only with the correct/shuffled memory; ranges reflect the shuffle controls. The reported p-value is the least favorable seed.

Correct-memory residuals carry sample-specific information in all three modalities. The vision probe further identifies pre-compression information weakly linearly decodable from the input summary; the protein and molecule probes identify memory directions weakly expressed in the decoder state before injection.

Table 25: Ablation on one-shot route stabilization components (Qwen-2.5-0.5B + SigLIP, $N_q = 128$, $\ell_r = 12$). All reported scores are in percentage points ($\times 100$). Perception/Reasoning grouping follows Appendix A.4; Overall is Norm-Avg (Appendix A.5). Best in **bold**, second-best underlined.

	Perception-dominant					*Reasoning-dominant*			*Norm-Avg*		
Setting	**GQA↑**	**MME-P↑**	**POPE↑**	**TextVQA↑**	**SQA^{I}↑**	**MME-C↑**	**MMMU↑**	**SQA↑**	**Perc.↑**	**Reas.↑**	**Overall↑**
ENCORE	57.04	**1288.30**	87.34	46.65	59.74	236.43	33.6	**60.43**	**84.36**	**79.45**	**82.52**
w/o entropy	56.80	1264.87	87.37	46.93	59.64	216.43	32.1	60.06	84.06	75.82	80.97
w/o normcap	**57.28**	1250.52	**87.91**	**47.37**	59.20	**237.14**	31.8	59.92	84.15	77.82	81.78
w/o gate	57.06	1240.87	87.40	47.21	**60.39**	226.07	**34.0**	**60.43**	84.11	78.63	82.06

Table 26: Sample retrieval from the recovery residual using the corresponding modality-memory target. Top-1/5 are percentages; MRR is on a 0–1 scale.

Modality	Memory	Top-1	Top-5	MRR
Vision	Correct	31.7	68.3	0.478
	Shuffled	5.0	17.5	0.154
Molecule	Correct	99.6	100.0	0.998
	Shuffled	1.1	9.2	0.081
Protein	Correct	69.5	96.5	0.805
	Shuffled	2.5	14.0	0.110

B.2.7 Recovery Depth under an Eight-Token Interface

We extend the vision depth sweep in Table 21 to the molecule setting, keeping $N_q = 8$ and all other training and stabilization settings fixed. Table 28 reports Overall at five recovery layers.

Layers 16 and 21 perform similarly, whereas layers 5 and 27 are less effective. The 5.18-point range is larger than the 2.17-point range in the vision sweep. This is consistent with greater sensitivity when the model relies more on recovery, while retaining a simple midpoint default across settings.

B.2.8 Reinjection Frequency Ablation

Setup. We study the effect of repeated decoder-side visual reinjection on Qwen-2.5-0.5B under the base ENCORE configuration. All settings are kept identical except for the reinjection layers: 1-shot (layer 12, default), 2-shot (layers 8, 16), 3-shot (layers 8, 12, 16), and 5-shot (layers 4, 8, 12, 16, 20). To isolate the effect of reinjection frequency from parameter count, all reinjection layers *share* a single set of route parameters.

Observations. (i) Increasing injection frequency does not monotonically improve performance: 2-shot and 3-shot both *decrease* the overall Norm-Avg relative to the 1-shot baseline (81.35 and 81.93 vs. 82.52), with 2-shot showing the largest reasoning drop (-2.51 on Reas. Norm-Avg).

(ii) 5-shot recovers and slightly surpasses 1-shot on the overall metric (82.73 vs. 82.52), with notable gains on SQA^{I} (+1.44) and SQA (+0.92), suggesting that sufficiently dense reinjection can benefit reasoning-intensive tasks.

(iii) The 5-shot configuration scores lower on MME-P (-38.22) and POPE (-0.47), while Perception Norm-Avg remains close (84.38 vs. 84.36). Given that the 1-shot configuration already captures

Table 27: Retrieval on four weakly linearly decodable memory directions, selected using the modality-specific protocol above. Top-1/5 are percentages.

Modality	Correct 1/5	Shuffled 1/5	C-only	S-only	Worst p
Vision	13.0/43.0	3.0/14.0	12	2	0.00647
Molecule	41.1/80.5	2.8/12.8	71–76	1	1.55×10^{-20}
Protein	14.4/43.1	2.9/17.9	21–22	2–5	0.00125

Table 28: Recovery depth in the molecule setting (Vicuna-v1.5-7B). The midpoint is used as the default.

Layer	Overall
5	84.68
11	88.64
16 (default)	89.82
21	89.86
27	86.63

Table 29: Ablation on decoder-side reinjection frequency (Qwen-2.5-0.5B + SigLIP, $N_q = 128$). "k-shot" denotes the number of reinjection layers; all variants share a single set of route parameters to isolate the effect of injection frequency from parameter count. Best in **bold**, second-best underlined.

	Perception-dominant					*Reasoning-dominant*			Norm-Avg		
Setting	**GQA↑**	**MME-P↑**	**POPE↑**	**TextVQA↑**	**SQA^{I}↑**	**MME-C↑**	**MMMU↑**	**SQA↑**	**Perc.↑**	**Reas.↑**	**Overall↑**
1-shot (ℓ=12)	57.04	**1288.30**	**87.34**	46.65	59.74	236.43	**33.6**	60.43	84.36	79.45	82.52
2-shot (8, 16)	**57.26**	1271.23	86.61	**47.93**	57.96	220.36	33.3	59.30	83.99	76.94	81.35
3-shot (8, 12, 16)	56.93	1249.85	87.32	47.24	59.64	**239.64**	32.2	60.08	83.98	78.49	81.93
5-shot (4, 8, 12, 16, 20)	57.16	1250.08	86.87	47.28	**61.18**	237.86	**33.6**	**61.35**	**84.38**	**80.00**	**82.73**

the majority of the dual-pathway benefit with a single injection point and the simplest deployment footprint, we adopt it as the default in all main experiments.

B.2.9 Local Hyperparameter Sensitivity

We vary one of $\kappa, \tau, \eta, \gamma_{\max}$ at a time on the same trained vision checkpoint and 128 validation examples. All eight perturbations retain the default macro accuracy of 63.021 and change zero final predictions. Table 30 reports the realized scale and relative hidden-state shift.

Table 30: One-factor changes to stabilization at inference. All rows use the same checkpoint and have identical final predictions on the 128-example subset.

Setting	Mean γ	Relative hidden shift
Default	0.09941	0.005668
$\kappa = 0.08$	0.09941	0.005650
$\kappa = 0.12$	0.09941	0.005668
$\tau = 0.14$	0.09957	0.005677
$\tau = 0.22$	0.09919	0.005655
$\eta = 6$	0.09792	0.005584
$\eta = 10$	0.09983	0.005692
$\gamma_{\max} = 0.08$	0.07953	0.004534
$\gamma_{\max} = 0.12$	0.11929	0.006801

Changing $\gamma_{\max}$ alters the update magnitude as expected while leaving predictions unchanged on this subset. The realized scale is close to its upper bound, which also explains the small changes induced by τ and η here.

B.3 Detailed Efficiency Analysis

To complement the main-text results in Section 3.2.4, we report detailed efficiency measurements across all three modalities with the original profiling setup. The additional H200 series is reported separately below. The token budgets are: vision $729 \to 128$, protein ~ 349 on average (range 18–1,023) $\to 64$, and molecule ~ 28 nodes on average (range 1–239) $\to 8$.

Complete efficiency tables. Table 31 reports per-sample FLOPs, throughput, and peak GPU memory for both training and inference. Table 32 reports stage-wise and total wall-clock training time.

Table 31: Training and inference efficiency comparison across modalities and LLM backbones. Backbone scale and connector token budgets (#Tok) are centralized in this table. FLOPs are per sample in TFLOPs (10^{12}). Throughput (Thru.) is in samples/second. Memory (Mem.) is peak GPU allocation in GiB.

			Training			Inference		
Modality	**LLM Backbone**	**Method (#Tok)**	**TFLOPs↓**	**Thru.↑**	**Mem.↓**	**TFLOPs↓**	**Thru.↑**	**Mem.↓**
Vision	Qwen-2.5-0.5B †	MLP (729)	5.32	5.72	31.21	1.65	2.67	11.98
		Q-Former (128)	3.06	5.16	29.82	0.92	2.76	11.94
		ENCORE (128)	3.34	5.60	29.37	1.01	2.65	11.79
	StableLM-2-1.6B ‡	MLP (729)	10.43	4.56	13.86	3.24	2.13	5.33
		Q-Former (128)	4.80	4.21	13.23	1.44	2.25	5.29
		ENCORE (128)	4.63	4.64	12.70	1.40	2.20	5.10
	Phi-2-2.7B‡	MLP (729)	17.29	3.32	20.72	5.37	1.55	7.97
		Q-Former (128)	6.89	3.18	20.88	2.07	1.70	8.35
		ENCORE (128)	6.65	3.48	20.04	2.01	1.65	8.05
Protein	Llama-3.1-8B-Instruct‡	MLP (∼349.15)§	17.76	3.8	20.5	11.88	0.65	18.35
		Q-Former (64)	7.32	8.41	22.07	4.91	0.40	17.19
		ENCORE (64)	7.27	8.24	22.56	4.85	0.36	17.25
Molecule	Vicuna-v1.5-7B†	MLP (∼28.23)	2.63	11.91	20.83	1.29	19.50	13.82
		Q-Former (8)	2.21	8.70	23.63	1.08	18.40	16.18
		ENCORE (8)	2.20	10.65	16.29	1.08	18.70	13.73

† Benchmarked on a 4-GPU cluster. ‡ Benchmarked on the NSCC cluster.

§ MLP requires halving the batch size to avoid OOM under the same GPU budget.

Table 33 reports end-to-end inference latency across all modality–backbone–benchmark settings, including data loading, encoding, and autoregressive decoding.

While per-sample FLOPs provide a useful hardware-agnostic proxy, measured runtime does not always follow FLOP trends. Wall-clock latency is also affected by GPU utilisation, memory-access patterns, operator regularity, and variable decoding lengths under autoregressive generation. We therefore interpret FLOPs, throughput, memory, and wall-clock time jointly.

Table 32: Wall-clock training time comparison across modalities and LLM backbones. All times are reported in `H:MM:SS` format.

Modality	**LLM Backbone**	**Method**	**Stage 1**	**Stage 2**	**Total**
Vision	Qwen-2.5-0.5B[1]	MLP	2:21:19	4:51:19	7:12:38
		Q-Former	1:42:44	4:41:56	6:24:40
		ENCORE	1:30:52	4:22:18	5:53:10
	StableLM-2-1.6B[2]	MLP	2:24:22	5:55:28	8:19:50
		Q-Former	1:36:12	4:58:27	6:34:39
		ENCORE	1:31:08	4:59:41	6:30:49
	Phi-2-2.7B[2]	MLP	7:50:16	19:15:36	27:05:52
		Q-Former	1:52:56	7:50:43	9:43:39
		ENCORE	1:48:12	7:55:25	9:43:37
Protein	Llama-3.1-8B[2]	MLP	11:03:00	112:38:00[3]	123:41:00
		Q-Former	29:18:36	54:12:00	83:30:36
		ENCORE	26:31:12	55:22:24	81:53:36
Molecule	Vicuna-v1.5-7B[1]	MLP	—	7:03:20	7:03:20
		Q-Former	—	11:12:48	11:12:48
		ENCORE	—	7:47:57	7:47:57

Key observations.

(i) Cost reduction scales with effective compression. Relative to MLP, ENCORE's FLOPs savings broadly follow the degree of token compression, consistent with Proposition 2. The reduction is largest on protein (training 59.1%, inference 59.2%) and vision–Phi-2 (training 61.5%, inference 62.6%), and smallest on molecule (16.3% in both phases), where the baseline sequence is already

Table 33: Wall-clock inference latency (seconds) across modalities, backbones, and benchmarks. All timings are measured end-to-end on the same hardware (the 4-GPU cluster). Best result per row is highlighted in **bold**; second-best is underlined.

Modality	Backbone	Benchmark	Task Type	#Samples	MLP	Q-Former	ENCORE
VISION	Qwen-2.5-0.5B	GQA	Perception / VQA	12578	**267**	282	268
		ScienceQA	Reasoning / QA	4241	149	139	**129**
		TextVQA	OCR / Perception	5000	283	284	**275**
		POPE	Perception	8910	345	347	**319**
		MME	Mixed	2374	98	99	**91**
		MMMU	Reasoning	900	77	74	**68**
	StableLM-2-1.6B	GQA	Perception / VQA	12578	404	310	**264**
		ScienceQA	Reasoning / QA	4241	219	150	**127**
		TextVQA	OCR / Perception	5000	382	300	**272**
		POPE	Perception	8910	594	420	**325**
		MME	Mixed	2374	149	120	**96**
		MMMU	Reasoning	900	288	200	**158**
	Phi-2-2.7B	GQA	Perception / VQA	12578	700	430	**399**
		ScienceQA	Reasoning / QA	4241	353	200	**178**
		TextVQA	OCR / Perception	5000	622	350	**313**
		POPE	Perception	8910	954	480	**413**
		MME	Mixed	2374	255	139	**120**
		MMMU	Reasoning	900	164	**101**	**101**
PROTEIN	Llama-3.1-8B	Prot2Text	Generation	4203	4075	2267	**2210**
		UniProtQA	QA	3000	2408	1425	**977**
		Protein Instructions	Instruction / QA	3000	1985	1039	**998**
MOLECULE	Vicuna-v1.5-7B	DrugBank	Generation	835	980	827	**807**
		ChEMBL	Classification	1372	185	192	**176**
		Cytotoxicity	Classification	570	121	129	**117**

short (∼28 tokens). Wall-clock training time follows the same pattern: ENCORE is faster than MLP in four of five settings, with savings ranging from 1.3 h (Vision–Qwen-2.5) to 41.8 h (Protein–Llama-3.1-8B). Molecule is the only exception, where MLP is 0.7 h faster because the fixed recovery overhead is not fully amortized.

(ii) ENCORE vs. Q-Former: similar arithmetic cost with modality-dependent trade-offs. ENCORE and Q-Former are generally close in FLOPs, but differ in practical efficiency depending on the modality. On molecule, ENCORE reduces training memory (16.29 vs. 23.63 GiB, 31.1% reduction) and improves throughput. On vision, ENCORE consistently improves training throughput across all backbones and typically reduces memory usage. On protein, the two methods remain close in FLOPs, with Q-Former slightly better in throughput and memory. For total wall-clock training time, ENCORE is never slower than Q-Former across the five settings, but the margin is highly uneven: it is substantial on molecule (7:47:57 vs. 11:12:48), yet negligible on Vision–Phi-2, where the difference is only 2 seconds. Overall, ENCORE should be viewed as a robust accuracy–efficiency trade-off option rather than a strict per-metric dominator.

(iii) System-level effects: FLOPs, throughput, and latency can diverge. The mismatch between FLOPs and measured runtime is most visible in protein inference: although ENCORE and Q-Former reduce FLOPs by 58.7–59.2% relative to MLP, MLP achieves higher throughput (0.65 vs. 0.36–0.40 samples/s). Operator regularity, memory access, utilization, and decoding length can all affect this comparison; the reported profiles do not isolate their contributions. Despite this, ENCORE achieves the lowest end-to-end inference latency in 23 out of 24 settings (Table 33), especially on larger backbones where reducing modality tokens lowers KV-cache size and attention cost. These results highlight that practical efficiency should be assessed at the system level rather than inferred from FLOPs alone.

Note on protein MLP. Under the same GPU budget, the protein MLP baseline requires halving the training batch size to avoid OOM. Protein sequences vary widely in length (18–1,023 tokens), and occasional long sequences inflate the full-token representation for the entire batch, causing memory spikes that exceed GPU capacity at the original batch size. The reported throughput and memory for protein MLP correspond to this reduced configuration, further illustrating the practical benefits of compression in high-cost settings.

Table 34: Inference-time TFLOPs and corresponding anchor-normalized average performance (Norm-Avg) for all methods in Table 31. Norm-Avg values are taken from the matched modality-specific result tables under the same settings.

Modality	LLM Backbone	Method	Inference TFLOPs↓	Norm-Avg↑
Vision	Qwen-2.5-0.5B	MLP	1.65	82.47
		Q-Former	0.92	74.29
		ENCORE	1.01	82.52
	StableLM-2-1.6B	MLP	3.24	85.68
		Q-Former	1.44	82.69
		ENCORE	1.40	86.53
	Phi-2-2.7B	MLP	5.37	94.19
		Q-Former	2.07	86.80
		ENCORE	2.01	94.37
Protein	Llama-3.1-8B-Instruct	MLP	11.88	89.58
		Q-Former	4.91	90.75
		ENCORE	4.85	93.34
Molecule	Vicuna-v1.5-7B	MLP	1.29	80.88
		Q-Former	1.08	82.66
		ENCORE	1.08	89.82

Inference cost–performance alignment. Table 34 lists the exact inference TFLOPs and Norm-Avg values underlying Figure 2; trade-off patterns are discussed in Section 3.2.4.

B.3.1 End-to-End Prefill, Decode, and Memory

We separate arithmetic cost from practical latency using the H200 profiling series described in Appendix B.1.3. Table 35 reports batch-size-one prefill latency, time to first token (TTFT), decode throughput, peak memory, and modality KV memory when available.

Table 35: End-to-end vision profiles at batch size one. Prefill and TTFT are milliseconds, decode is tokens/s, peak memory is GB, and modality KV is MB. A dash indicates an unreported quantity. Scores, token counts, and FLOPs appear in Table 15.

Method	Prefill	TTFT	Decode	Peak mem.	Modality KV
Qwen-2.5-0.5B					
MLP	14.51	27.09	85.76	4.53	—
Q-Former	13.48	33.06	83.37	4.27	—
Adaptive-pool	14.11	26.91	85.28	4.02	—
C-Abstractor	14.38	28.56	83.49	4.06	—
TokenPacker	14.03	27.35	83.76	4.04	—
FastV-FT (128)	14.34	26.99	82.62	4.01	—
ViCA	14.62	29.21	84.36	4.57	—
ENCORE	14.61	27.06	84.19	4.07	—
Qwen-2.5-7B					
MLP	34.13	47.22	58.82	17.58	39.82
ViCA	34.32	47.74	60.73	17.73	40.13
Q-Former	19.54	38.81	59.21	17.22	7.03
Adaptive-pool	21.18	35.86	58.96	17.10	7.91
C-Abstractor	21.41	36.33	57.88	17.14	7.91
TokenPacker	21.09	35.63	58.52	17.12	7.91
FastV (364)	24.21	37.43	55.42	17.25	22.81
FastV (128)	20.53	33.62	55.23	17.06	11.84
FastV-FT (128)	20.53	33.71	55.23	17.06	11.84
ENCORE	20.91	35.24	59.74	17.14	7.03

At 0.5B, ENCORE has similar prefill and TTFT to MLP, despite fewer FLOPs; the peak-memory reduction is more visible. At 7B, ENCORE and ViCA remain close in Norm-Avg, while ENCORE's prefill, TTFT, and modality KV memory are 60.9%, 73.8%, and 17.5% of ViCA's. Batch-size-one decode throughput remains similar, since recovery is invoked once during prefill and does not introduce a memory read at each generated token.

Backbone scaling. Table 36 extends the ENCORE/ViCA comparison across the four vision backbones. Decode throughput is a speed ratio (higher is better); the remaining columns are cost ratios (lower is better).

Table 36: ENCORE/ViCA profiling ratios, in percent. These ratios belong to the end-to-end series.

Backbone	Infer	Prefill	TTFT	Decode	Peak	KV
Qwen-2.5-0.5B	61.2	99.9	92.6	99.8	89.1	17.6
StableLM-2-1.6B	40.0	96.8	91.9	99.3	96.3	17.6
Phi-2-2.7B	34.4	67.2	81.1	99.6	95.8	17.6
Qwen-2.5-7B	33.1	60.9	73.8	98.4	96.7	17.5

B.3.2 Batched Inference

The companion batch sweep measures total decode throughput for MLP, ViCA, and ENCORE (Table 37). At batch size 128, ENCORE reaches 2,562 tokens/s versus 1,078 for ViCA, and it remains executable at batch size 256.

Table 37: Batched decode throughput (tokens/s). OOM denotes a run exceeding the memory capacity of the profiling setup.

Method	$B = 1$	$B = 8$	$B = 32$	$B = 64$	$B = 128$	$B = 256$
MLP	59	392	799	979	1077	OOM
ViCA	60	397	802	981	1078	OOM
ENCORE	60	415	1400	2065	2562	2906

B.3.3 Isolated Recovery Cost

We profile one recovery invocation on a single H200 in fp16, with batch size one and $T_t = 64$. Table 38 reports the route's arithmetic cost, latency, and temporary memory across six modality–backbone settings. These measurements cover the one-time recovery operation and its temporary memory.

Table 38: Isolated recovery invocation. Arithmetic cost is measured in GFLOPs, latency in milliseconds, and temporary memory in MiB.

Modality	Backbone	GFLOPs	Latency	Memory
Vision	Qwen-2.5-0.5B	4.23	0.568	8.04
Vision	StableLM-2-1.6B	17.13	0.545	17.56
Vision	Phi-2-2.7B	25.57	0.556	21.95
Vision	Qwen-2.5-7B	47.42	0.561	30.73
Protein	Llama-3.1-8B	8.67	0.506	5.06
Molecule	Vicuna-v1.5-7B	4.85	0.500	3.70

The measured one-time latency is 0.500–0.568 ms and temporary memory is 3.70–30.73 MiB. Protein cost depends on actual sequence length: the benchmark averages about 349 encoder tokens, below the fixed 729-token vision input.

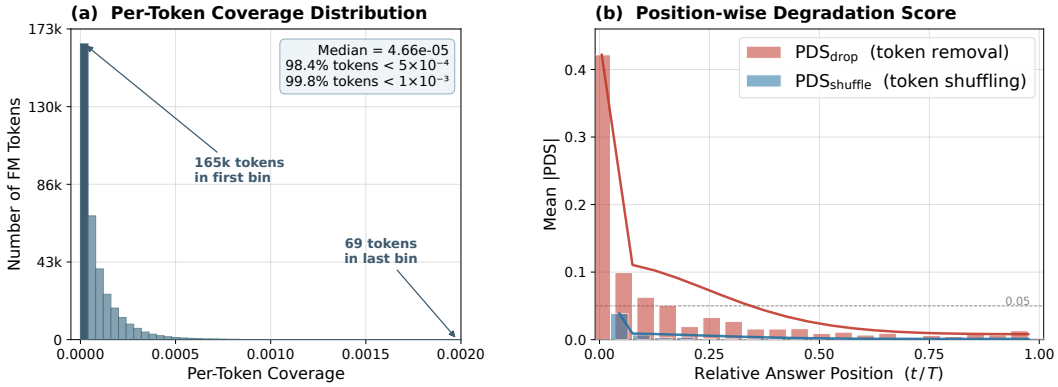

Figure 4: **Sparse and temporally localized use of FM-token information in the protein MLLM. (a)** PTC distribution across FM tokens. The vast majority of tokens receive near-zero answer-side attention, indicating that only a small subset is substantially attended to during decoding. **(b)** Mean PDS after dropping or shuffling FM tokens, binned by relative answer position. Dropping tokens causes the largest degradation at early answer positions, with diminishing impact thereafter. Shuffling yields substantially smaller degradation than dropping, suggesting lower sensitivity to token order than to token removal in this setting.

B.4 Task-Specific and OOD Analyses

B.4.1 FM-Token Use in the Protein Interface

We examine how a protein MLLM uses encoder-derived prefix tokens through two diagnostic probes: attention coverage and sensitivity to token perturbations.

Protocol. We analyze FM-token redundancy under teacher forcing on $N{=}256$ randomly sampled training examples (seed 42). For each example, let F denote the set of FM-token positions in the input prefix and A the set of answer-token positions with valid supervision labels. All probabilities are computed using gold answer tokens. The FM-token definition is modality-specific: protein uses adapter-projected sequence embeddings, vision uses image-side prefix embeddings indicated by the visual token mask, and molecule uses graph-derived prefix embeddings indicated by the molecule token mask.

Per-Token Coverage (PTC). For each FM token j, we compute the average attention it receives from answer tokens across all layers and heads:

$$\mathrm{PTC}_j = \frac{1}{L \cdot H \cdot |A|} \sum_{l=1}^{L} \sum_{h=1}^{H} \sum_{t \in A} \alpha_{t \to j}^{(l,h)},$$

where $\alpha_{t \to j}^{(l,h)}$ is the attention weight from answer token t to FM token j at layer l, head h. A PTC value near zero indicates that the token receives negligible answer-side attention.

Position-wise Degradation Score (PDS). We measure how FM-token perturbations affect answer prediction at different generation stages. For each answer position t with gold token y_t:

$$\mathrm{PDS}(t) = \log p_{\mathrm{orig}}(y_t) - \log p_{\mathrm{perturb}}(y_t),$$

where p_{orig} is the model distribution under original FM tokens and p_{perturb} is computed after perturbation. We consider two perturbation types: (i) *Drop*: replace $K = \mathrm{round}(0.9 \cdot |F|)$ randomly selected FM-token embeddings with zero vectors, keeping sequence length unchanged (at least one FM token is always retained); (ii) *Shuffle*: apply a full random permutation over all FM-token positions, preserving token identity but disrupting order. To visualize positional trends, we normalize answer positions to $[0, 1]$ and aggregate PDS into $B{=}20$ equal-width bins, reporting mean absolute PDS per bin. All perturbations are deterministic given a per-sample seed derived from the global seed.

Results. Figure 4(a) shows that the PTC distribution is sharply concentrated near zero: most FM tokens receive negligible answer-side attention, while only a small subset is attended more substantially. Attention concentration suggests uneven use of the FM-token interface.

Figure 4(b) reveals two complementary findings. First, dropping FM tokens causes the largest degradation at the earliest answer positions, while the impact on later positions is much smaller, suggesting that FM information is consumed primarily during the initial decoding phase and that the model increasingly relies on its own language prior thereafter. Second, shuffling FM tokens produces substantially less degradation than dropping them, indicating that the model depends more on which tokens are present than on their sequential arrangement.

Together, these probes suggest concentrated, early use of protein prefix information, motivating a compact persistent interface.

B.4.2 Protein-Modal OOD Benchmarks

To further evaluate out-of-distribution generalization in the protein setting, we test on two external benchmarks that differ from the in-domain Prot2Text evaluation in supervision format and query distribution. The first is the protein-oriented subset of Mol-Instructions (Fang et al., 2024), a large-scale biomolecular instruction-tuning corpus. The second is UniProtQA from BioMedGPT (Luo et al., 2024), a protein question-answering benchmark designed for biomedical reasoning. These two benchmarks provide complementary distribution shifts in instruction style and QA structure, making them suitable OOD probes.

As shown in Table 39, ENCORE achieves the strongest overall profile on Mol-Instructions across both generation and retrieval metrics. On UniProtQA, ENCORE also leads most metrics while Q-Former remains best on exact match. These results extend the aggregate performance comparison beyond the primary training distribution.

Table 39: Results on protein-modal OOD benchmarks (Mol-Instructions and UniProtQA). All scores are reported in percentage points ($\times 100$). RBT and BBT denote bidirectional retrieval F1. Norm-Avg follows the anchor-normalized definition in Appendix A.5. Best in **bold**, second-best underlined.

Dataset	Method	*Text Generation Metrics*						*Retrieval Metrics*		*Overall*
		E-Match ↑	BLEU-2 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	RBT ↑	BBT ↑	Norm-Avg ↑
Mol-Instructions	MLP	0.00	27.72	25.88	49.98	43.50	47.89	89.76	85.29	88.41
	Q-Former	0.00	**33.08**	30.45	52.71	45.28	49.95	90.29	86.20	94.41
	ENCORE	0.00	32.98	**30.79**	**53.70**	**47.31**	**51.49**	**90.54**	**86.77**	**96.06**
UniProtQA	MLP	11.10	16.87	14.47	20.63	13.10	19.44	84.92	77.44	55.67
	Q-Former	**12.23**	16.79	14.39	21.67	15.31	20.87	85.03	77.66	57.60
	ENCORE	11.77	**18.04**	**15.48**	**22.32**	**15.37**	**21.07**	**85.05**	**77.98**	**58.62**

C Theoretical Foundations of Compression and Recovery in ENCORE

This appendix provides supporting analysis for two aspects of ENCORE's design. First, we derive when token compression yields net computational savings over a token-preserving baseline (Appendix C.2). Second, we give a local differential analysis of how compression suppresses certain task-relevant input directions and how direct feature injection can partially restore sensitivity to them (Appendix C.3). The two parts are complementary: the first justifies the efficiency gain, while the second clarifies the representational cost that the recovery route is designed to mitigate.

C.1 Unified Notation Reference

Table 40 summarizes notation shared between Section 2 and this appendix.

Here g_{cmp} abstracts the compression pipeline in Section 2, and S abstracts the effective first-order contribution of the recovery route.

Table 40: Unified notation used in the method and theory sections.

Symbol	Meaning
x_m	Modality input instance.
$x = \mathrm{vec}(V)$	Vectorized encoder representation used in local analysis.
$\mathcal{E}$ / $\mathcal{F}$	Pretrained modality encoder / pretrained decoder-only LLM.
$V \in \mathbb{R}^{N_v \times d_v}$	Raw modality features produced by $\mathcal{E}$.
$V_m \in \mathbb{R}^{N_v \times d}$	Full modality memory used by the recovery route.
$M_q \in \mathbb{R}^{N_q \times d}$	Compressed modality tokens.
N_v, N_q, T_t	Raw modality tokens, compressed tokens, and text tokens.
N_{base}, N_{enc}	Sequence lengths before and after compression.
L, ℓ_r	Decoder depth and recovery layer index.
$g_{\mathrm{mlp}}, g_{\mathrm{cmp}}$	Token-preserving and compressive mappings.
$D_{\mathrm{mlp}} = N_v d$, $D_{\mathrm{cmp}} = N_q d$	Interface dimensions.
f	Downstream task map/readout applied to the interface representation.
S	Effective recovery mapping under local linearization.

C.2 Computational Complexity Analysis

For a decoder layer with sequence length N and hidden size d, we use[1]

$$f_{\mathrm{fwd}}(N) = 24Nd^2 + 4N^2 d.$$

The baseline and compressed costs are:

$$C^{(b)}_{\mathrm{tot}} = C^{(b)}_{\mathrm{proj}} + L f_{\mathrm{fwd}}(N_{\mathrm{base}}), \quad C^{(c)}_{\mathrm{tot}} = C^{(c)}_{\mathrm{proj}} + L f_{\mathrm{fwd}}(N_{\mathrm{enc}}),$$

where $C^{(b)}_{\mathrm{proj}}$ and $C^{(c)}_{\mathrm{proj}}$ denote the total depth-independent prefill FLOPs of the token-preserving and compressed designs, respectively. For the compressed design, this includes the embedding-level compressor and the recovery route.

Let $\Delta N = N_v - N_q > 0$. Then

$$\Delta f_{\mathrm{fwd}} = 24\,\Delta N\, d^2 + 4\,\Delta N\,(2T_t + N_v + N_q)\, d. \tag{14}$$

Since $\Delta N > 0$, $d > 0$, and $T_t, N_v, N_q \geq 0$, we have $\Delta f_{\mathrm{fwd}} > 0$.

Proposition 2 (Depth-scaled efficiency advantage)**.** *Assume $\Delta N = N_v - N_q > 0$. Define*

$$L^* := \frac{C^{(c)}_{\mathrm{proj}} - C^{(b)}_{\mathrm{proj}}}{\Delta f_{\mathrm{fwd}}}.$$

Then $C^{(c)}_{\mathrm{tot}} < C^{(b)}_{\mathrm{tot}}$ if $L > L^*$. In particular, if $C^{(c)}_{\mathrm{proj}} \leq C^{(b)}_{\mathrm{proj}}$, then $L^* \leq 0$ and the compressed design is cheaper at every positive depth.

Proof. Since $\Delta f_{\mathrm{fwd}} > 0$, rearranging $C^{(c)}_{\mathrm{tot}} < C^{(b)}_{\mathrm{tot}}$ gives $C^{(c)}_{\mathrm{proj}} - C^{(b)}_{\mathrm{proj}} < L\,\Delta f_{\mathrm{fwd}}$, which holds whenever $L > L^*$. When $C^{(c)}_{\mathrm{proj}} \leq C^{(b)}_{\mathrm{proj}}$, the numerator is non-positive, so $L^* \leq 0$ and the inequality is satisfied for all $L \geq 1$. □

C.3 Local Differential View of Compression and Recovery

C.3.1 Information-theoretic bound

This formalizes Proposition 1(i). Here $I(\cdot;\cdot)$ denotes mutual information.

Condition on a fixed instruction throughout this analysis. Let

$$x = \mathrm{vec}(V) \in \mathbb{R}^{D_{\mathrm{in}}}, \quad D_{\mathrm{in}} = N_v d_v.$$

[1]The constant 24 assumes a standard FFN with expansion ratio 4, i.e. two projections of size $d \times 4d$ and $4d \times d$. Architectures using a different expansion ratio change this constant but do not affect the existence of the depth threshold L^*.

Since $M_q = g_{\text{cmp}}(x)$ is deterministic, the data processing inequality gives

$$I(x;\, f(M_q)) \le I(x;\, M_q). \tag{15}$$

That is, once information is discarded by the compressed interface g_{cmp}, it cannot be reconstructed from M_q alone by any deterministic downstream map applied to M_q.

C.3.2 Local differential bound on the null-space gap

We now switch from a global information-theoretic view to a local differential perspective. This formalizes Proposition 1(ii).

By rank–nullity applied to the Jacobian of any smooth map $g\colon \mathbb{R}^{D_{\text{in}}} \to \mathbb{R}^{D_{\text{out}}}$,

$$\dim\ker(J_g(x)) \ge D_{\text{in}} - D_{\text{out}}. \tag{16}$$

Null-space gap under compression. If g_{mlp} operates at full output rank, i.e. $\operatorname{rank}(J_{g_{\text{mlp}}}(x)) = D_{\text{mlp}}$, then $\dim\ker(J_{g_{\text{mlp}}}(x)) = D_{\text{in}} - D_{\text{mlp}}$, and rank–nullity for the compressive map gives

$$\dim\ker(J_{g_{\text{cmp}}}(x)) - \dim\ker(J_{g_{\text{mlp}}}(x)) \ge (D_{\text{mlp}} - D_{\text{cmp}}) = (N_v - N_q)\, d. \tag{17}$$

Full output rank requires $d_v \ge d$; it is not assumed for all backbone dimensions. Without that idealization, the same argument gives a gap of at least $\operatorname{rank}(J_{g_{\text{mlp}}}) - N_q d$. Section C.4 examines the primary vision configuration.

C.3.3 Task-relevance assumption

The null-space gap above is a structural consequence of dimensional compression. By itself, it does not imply downstream degradation; this depends on whether the suppressed directions are relevant to the downstream task.

Assumption 3 (Task-relevant suppressed directions)**.** *There exists* $u \in \ker(J_{g_{\text{cmp}}}(x)) \setminus \ker(J_{g_{\text{mlp}}}(x))$ *such that* $J_f(g_{\text{mlp}}(x))J_{g_{\text{mlp}}}(x)u \neq 0$.

The assumption concerns local task sensitivity and is not directly established by aggregate benchmark scores. Prior compression studies report task-dependent degradation (Peng et al., 2025; Yang et al., 2024; Ye et al., 2025). Our matched recovery controls (Appendix B.2.3) and probes of information weakly decodable from the summary (Appendix B.2.6) provide indirect evidence consistent with a task-relevant bottleneck. They do not enumerate the null space or prove that every suppressed direction matters. Likewise, a residual performance gap does not by itself identify an unrecoverable subspace.

C.3.4 First-order distinguishability

Proposition 4 (First-order distinguishability gap)**.** *Under Assumption 3,*

$$g_{\text{cmp}}(x + \epsilon u) = g_{\text{cmp}}(x) + o(|\epsilon|),$$

whereas

$$g_{\text{mlp}}(x + \epsilon u) = g_{\text{mlp}}(x) + \epsilon J_{g_{\text{mlp}}}(x)u + o(|\epsilon|), \qquad J_{g_{\text{mlp}}}(x)u \neq 0.$$

Proof. By first-order Taylor expansion of g_{cmp} around x,

$$g_{\text{cmp}}(x + \epsilon u) = g_{\text{cmp}}(x) + \epsilon \underbrace{J_{g_{\text{cmp}}}(x)\, u}_{=0} + o(|\epsilon|) = g_{\text{cmp}}(x) + o(|\epsilon|),$$

where the linear term vanishes because $u \in \ker(J_{g_{\text{cmp}}}(x))$. For the token-preserving map, the same expansion gives

$$g_{\text{mlp}}(x + \epsilon u) = g_{\text{mlp}}(x) + \epsilon\, J_{g_{\text{mlp}}}(x)\, u + o(|\epsilon|).$$

Since $u \notin \ker(J_{g_{\text{mlp}}}(x))$ by construction, the linear term satisfies $J_{g_{\text{mlp}}}(x)\, u \neq 0$, so g_{mlp} remains first-order sensitive to the perturbation ϵu. □

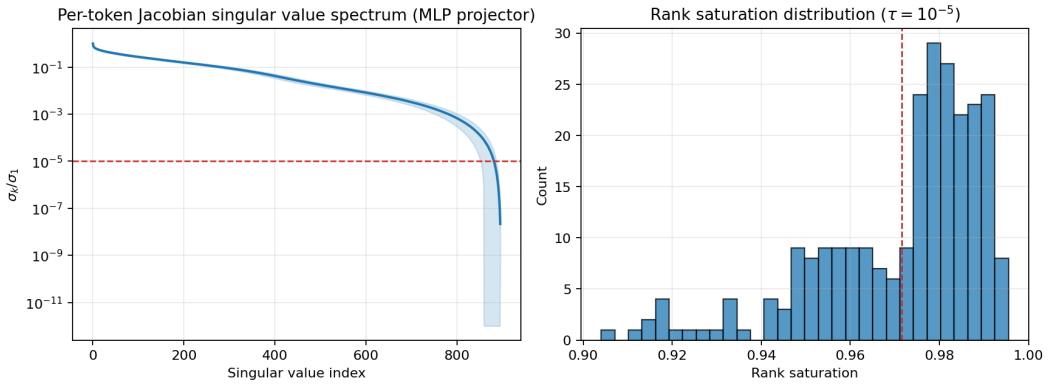

Figure 5: Empirical Jacobian rank analysis of the token-wise MLP projector. **Left:** Average normalized singular-value spectrum σ_k/σ_1 across 256 sampled tokens (shaded: ± 1 s.d.); the dashed line marks the numerical-rank threshold $\tau_{\text{rank}} = 10^{-5}$ **Right:** Distribution of per-token rank saturation $\text{rank}(J_i)/\min(d, d_v)$; the dashed line marks the mean (0.972). Even the lowest observed saturation (0.904) leaves the aggregate MLP rank more than $5\times$ larger than the compressed interface's rank upper bound.

C.3.5 Recovery via injection

Let $S\colon \mathbb{R}^{N_v d_v} \to \mathbb{R}^{T_t \times d}$ denote the effective first-order map from raw features x to the recovery residual at layer ℓ_r, induced by the cross-attention path $x \mapsto V_m \mapsto \text{CrossAttn}(H^{\text{text}}_{\ell_r}, V_m)$. Crucially, this path is *not* factored through g_{cmp}: S reads the full modality memory V_m directly, bypassing the compressed summary M_q. This architectural independence is what makes recovery possible at first order. Were S to receive only M_q, the chain rule would force $J_S(x) = J_{S'}(M_q)\, J_{g_{\text{cmp}}}(x)$ for some S', so any $u \in \ker(J_{g_{\text{cmp}}}(x))$ would automatically satisfy $J_S(x)\, u = 0$, making recovery of suppressed directions impossible. The following proposition formalizes this argument.

Proposition 5 (Recovery via injection)**.** *Let* $H^{\text{text}}_{\ell_r} = \psi(g_{\text{cmp}}(x))$ *and*

$$\tilde{H}^{\text{text}}_{\ell_r} = H^{\text{text}}_{\ell_r} + \alpha\, S(x).$$

For any $u \in \ker(J_{g_{\text{cmp}}}(x))$*, if* $\alpha \neq 0$ *and* $J_S(x)\, u \neq 0$*, then* $J_{\tilde{H}^{\text{text}}_{\ell_r}}(x)\, u \neq 0$.

Proof. By the chain rule, $J_{H^{\text{text}}_{\ell_r}}(x) = J_\psi(g_{\text{cmp}}(x))\, J_{g_{\text{cmp}}}(x)$. Since $u \in \ker(J_{g_{\text{cmp}}}(x))$, we have $J_{H^{\text{text}}_{\ell_r}}(x)\, u = 0$. Therefore

$$J_{\tilde{H}^{\text{text}}_{\ell_r}}(x)\, u = \underbrace{J_{H^{\text{text}}_{\ell_r}}(x)\, u}_{=0} + \alpha\, J_S(x)\, u = \alpha\, J_S(x)\, u \neq 0. \qquad \square$$

Remark 6 (Scope of recovery). Recovery is partial: directions in $\ker(J_{g_{\text{cmp}}}) \cap \ker(J_S)$ remain unrecoverable. In practice, S is induced by a single cross-attention-based recovery route with only T_t text query positions reading from N_v memory tokens. This limited query budget constrains how many suppressed directions can be reintroduced simultaneously, so the unrecoverable subspace can still be non-trivial, especially when $T_t \ll N_v$.

C.4 Empirical Evidence for Near-Maximal Per-Token Rank

The null-space comparison in Appendix C.3.2 requires near-maximal Jacobian rank for the token-wise MLP projector. Because the MLP Jacobian is block-diagonal across tokens, we verify rank saturation at the per-token block level.

We compute per-token Jacobians for 256 randomly sampled (example, token-position) pairs using a numerical-rank threshold $\sigma_k/\sigma_1 > 10^{-5}$. Figure 5 summarizes the results.

Results. Per-token rank saturation $\text{rank}(J_i)/\min(d, d_v)$ is high (mean 0.972, median 0.978, minimum 0.904), with numerical rank mean 870.6 out of 896.

Algorithm 1 ENCORE full forward pass (prefill at inference)

Input Modality input x_m; text tokens $\{t_1, \ldots, t_{T_t}\}$; prompt-index set $\mathcal{T}_{\text{prompt}}$; pretrained modality encoder $\mathcal{E}$; pretrained LLM $\mathcal{F}$ with L layers; injection index $\ell_r = \lfloor L/2 \rfloor$
Embedding-level compression
$V \leftarrow \mathcal{E}(x_m)$
$V_m \leftarrow VW_m$
$s \leftarrow \text{MeanPool}\big(\{\text{Emb}(t_i) \mid i \in \mathcal{T}_{\text{prompt}}\}\big)$
$\Delta q \leftarrow (sW_A)W_B/\sqrt{r_q}$
$Q \leftarrow Q_0 + \mathbf{1}_{N_q}\Delta q$
$Q_v \leftarrow \text{FFN}\Big(\text{CrossAttn}\big(\text{SelfAttn}(Q), V\big)\Big)$
$M_q \leftarrow \text{LN}(Q_v)W_o$
$H_0 \leftarrow \text{Concat}(M_q,\ \text{Emb}(t_1), \ldots, \text{Emb}(t_{T_t}))$
Decoder forward with single-Layer recovery injection
for $l = 1$ **to** L **do**
    $H_l \leftarrow \text{DecoderLayer}_l(H_{l-1})$
    if $l = \ell_r$ **then**
        Split H_l into $[H_l^{\text{mod}};\ H_l^{\text{text}}]$
        $Z_{\ell_r} \leftarrow \text{CrossAttn}(H_l^{\text{text}}, V_m)$
        Compute e_{ℓ_r} via Eq. 11, then γ_{ℓ_r} via Eq. 12
        Compute Δ_{ℓ_r} by Eq. (10)
        $\tilde{H}_l^{\text{text}} \leftarrow H_l^{\text{text}} + \gamma_{\ell_r}\,\Delta_{\ell_r}$
        $H_l \leftarrow \text{Concat}(H_l^{\text{mod}},\ \tilde{H}_l^{\text{text}})$
    end if
end for
$\mathbf{y} \leftarrow \text{LMHead}(H_L)$
Output Logits $\mathbf{y}$

Implication. In the vision setting ($N_v = 729$, $N_q = 128$, $d = 896$), the estimated aggregate MLP rank is

$$\widehat{\text{rank}}(J_{g_{\text{mlp}}}) \approx 729 \times 870.6 \approx 634{,}600,$$

whereas the compressed rank upper bound is

$$D_{\text{cmp}} = N_q d = 128 \times 896 = 114{,}688.$$

This yields a $5.5\times$ margin. These measurements provide empirical evidence that the token-wise MLP projector operates near maximal rank on sampled vision tokens, supporting the plausibility of the large null-space gap induced by compression in practice.

Relationship to the full-rank condition. The numerical saturation of 0.972 supports a large rank gap in this sampled vision configuration, not exact full output rank. Equation (17) is a conditional lower bound under its stated assumption; it is not an empirical upper bound. For measured rank below full output rank, the appropriate lower bound is $\text{rank}(J_{g_{\text{mlp}}}) - N_q d$ as described above.

D Full Forward-Pass Procedure

Algorithm 1 consolidates the complete ENCORE forward pass, covering embedding-level compression and a single recovery update at layer ℓ_r. At inference this procedure is run during prefill; later decode steps use the decoder cache and skip the recovery route. All notation follows Section 2.

E Expanded Related Work

Full-Token Interfaces. LLaVA-style connectors project modality features into the LLM embedding space and concatenate them with text tokens (Liu et al., 2023a; 2024a). While simple and modular, this design retains all modality tokens throughout the decoder, with self-attention cost scaling quadratically with the combined sequence length.

Compressed Interfaces and Token Reduction. To improve efficiency, query-based connectors compress modality features into a small set of latent tokens through cross-attention (Li et al., 2023a; Jaegle et al., 2021; Alayrac et al., 2022; Dai et al., 2023). mPLUG-Owl similarly uses a visual abstractor to construct a compact interface (Ye et al., 2024). Alternative designs include locality-aware projection in Honeybee, adaptive pooling in DeCo, and coarse-to-fine feature aggregation in TokenPacker (Cha et al., 2024; Yao et al., 2024; Li et al., 2025). Token pruning instead reduces the visual tokens retained during decoder computation, as in FastV (Chen et al., 2024); DualSpeed extends pruning to efficient training while addressing the mismatch between pruned training and full-token inference (Zhang et al., 2026). These approaches reduce computation, but aggressive token reduction can discard fine-grained modality information (Yang et al., 2025; Tan et al., 2025; Wang et al., 2026a).

Decoder-Side Fusion. Complementary approaches introduce modality information inside the decoder through cross-attention, gated adapters, or modality-specific processing (Alayrac et al., 2022; Gao et al., 2023; Wang et al., 2024; Ye et al., 2023). ViCA lets visual tokens bypass decoder self-attention and feed-forward layers, interacting with text through cross-attention at selected layers (Liu et al., 2026). DeepStack distributes visual features across depth through position-aligned residual addition (Meng et al., 2024). These designs motivate examining both where modality information enters the decoder and how it is integrated.

Connector Taxonomies and Efficient MLLMs. Connector-focused reviews categorize MLLM connectors by mapping, compression, mixture-of-experts, and multi-layer interaction patterns, highlighting connector design as a key factor in the performance–efficiency trade-off (Zhu et al., 2025). More broadly, efficient-MLLM surveys place connector compression alongside token reduction, lightweight backbones, and inference optimization as major routes toward practical multimodal systems (Jin et al., 2025).

Scientific Multimodal Extensions. Connecting modality encoders to LLMs is also central to scientific multimodal systems. In proteins, ProteinGPT combines sequence and structure encoders with projection layers and an LLM (Xiao et al., 2025); BioBridge, Prot2Chat, Prot2Text-V2, and ProtChatGPT further explore protein–language alignment and modeling (Wang et al., 2026b; 2025b; Fei et al., 2025; Wang et al., 2025a). In molecular modeling, GIT-Mol integrates graph, image, and text inputs, MolCA studies molecular graph–language alignment, and DrugChat explores multimodal drug modeling (Liu et al., 2024b; 2023b; Liang et al., 2024). These systems motivate studying connector design across modality families.

Relation to the Present Work. ENCORE combines instruction-conditioned input compression with one text-conditioned recovery read and stabilized residual integration. The information-bottleneck analysis motivates access beyond the input summary, while the component studies examine the value of the complete route. Table 11 compares the mechanisms, and Appendix B.2 reports the corresponding controls.

F Broader Impact

ENCORE improves the efficiency of multimodal large language models by reducing token-level computation while preserving conditioning quality. This can lower the computational and energy cost of deploying multimodal systems, making them more accessible in resource-constrained settings. Improved reasoning over scientific modalities, such as proteins and molecules, may also support downstream applications in biological research and early-stage drug discovery.

However, ENCORE inherits the risks of the pretrained encoders, language models, and datasets on which it is built. Biases, annotation errors, or distribution shifts in these components may lead to unreliable or misleading outputs, especially in high-stakes scientific or biomedical settings. ENCORE should therefore be viewed as a research tool rather than a standalone decision-making system for clinical, safety-critical, or regulatory use. Further work is needed to improve robustness, interpretability, and responsible deployment.