Rethinking Projector Training in Multimodal LLMs

Fabian Gröger[1,2,3,*], **Naël Ouerghemi**[1,*], **Shuo Wen**[1,*], **Maria Brbić**[1]
[1]EPFL [2]University of Basel [3]HSLU

Abstract

Multimodal LLMs are typically trained by first aligning a vision encoder with a language model via projector training on large image-caption datasets, followed by joint training of the full multimodal model. Despite its widespread use, the role of the projector training stage remains poorly understood. We find that *(i)* using only 10 to 20% of the data for training the projector already recovers most downstream performance gains, and *(ii)* projectors trained on different datasets can be linearly interpolated while largely preserving performance. These results suggest that projector training mainly performs a coarse alignment, which many different projectors achieve equally well. Motivated by these findings, we propose PORTAL, a training-free projector initialization that requires *neither paired image-caption data nor gradient-based optimization*, but only per-modality summary statistics. PORTAL computes an optimal-transport-based map between Gaussian approximations of the vision and language feature distributions on a shared principal subspace, so that the projector training stage can be skipped entirely. Without any paired data, PORTAL matches the standard pipeline trained on 558k image-caption pairs across six LLM backbones, two vision encoders, and 16 benchmarks. It outperforms that pipeline in 3 of 7 (vision encoder, LLM) settings and reaches an average 98% of its score across the other four.

1 Introduction

Modern multimodal LLMs commonly align a vision encoder with a language model by first training a projector on a large amount of image-caption paired samples, followed by joint training of the full multimodal model. This pipeline has become standard, from LLaVA [1, 2], Molmo [3], and InternVL [4] to the QwenVL series [5]. While the first stage of projector alignment has been motivated by the goal of producing a visual tokenizer compatible with the frozen LLM [1], what this stage *actually* does remains poorly understood. Prior works offer conflicting evidence: Prismatic [6] reports that skipping projector alignment can improve performance, whereas Cambrian-1 [7] shows that scaling the training data of projector alignment to 2.5M image-caption pairs helps. MM1 [8] suggests that factors like resolution and data mixing matter more than the projector design. These findings are difficult to reconcile without a deeper understanding of what projector alignment actually does.

In this work, we show that projector alignment primarily performs *coarse alignment*, mapping visual features into a language-model-compatible input space which admits a large set of possible solutions. Two empirical observations support this view: *(i)* using only 10 to 20% of the standard 558k-sample alignment corpus achieves similar performance to training on the full dataset, and *(ii)* projectors trained on entirely different datasets can be linearly interpolated without losing performance. If the projector alignment learned a precise and semantically unique mapping, neither reducing the training data nor linear interpolation between independently trained projectors would preserve performance.

Motivated by these observations, we propose PORTAL (**P**rojector via **O**ptimal-transpo**RT** **AL**ignment), a training-free approach that constructs the projector from the marginal statistics

[*]Joint first authorship

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

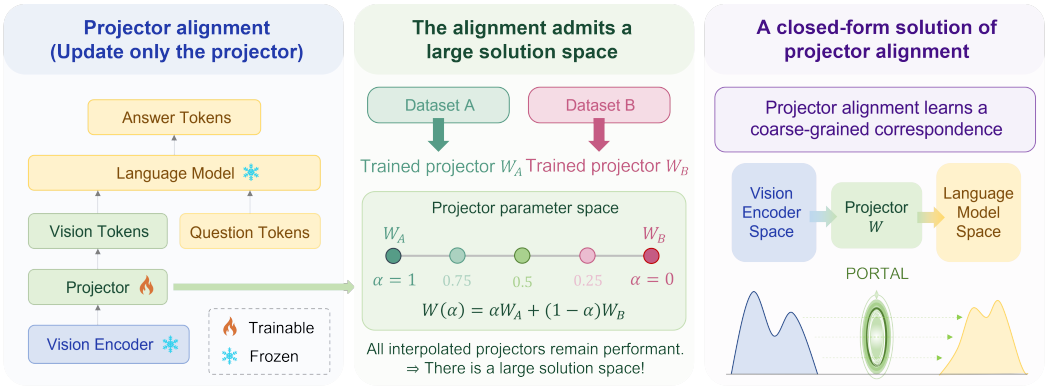


Figure 1: **Projector alignment as a coarse alignment with a broad solution space.** (*Left*) Standard pipeline: during projector alignment, the vision encoder and language model are frozen, and only the projector is trained to map visual tokens into a representation space compatible with the LLM using image-caption pairs. (*Middle*) We show that projectors trained on different image-caption datasets can be linearly interpolated while largely preserving downstream performance, indicating that there is a large solution space for the weights of the projector. (*Right*) This observation motivates a closed-form solution of projector alignment. We propose a training-free projector PORTAL constructed by optimal transport between the two modalities' marginal statistics.

of the two modalities. PORTAL is the Bures-Wasserstein optimal transport map between Gaussian approximations of the vision feature distribution and the language-model input distribution, computed in a shared top-k principal subspace. It does not require *any paired captions or gradient-based optimization*, relying only on the per-modality means, principal directions, and the spread along each direction. PORTAL removes the projector alignment training stage entirely.

Across seven (vision encoder, LLM) pairs spanning four LLM families (Vicuna, LLaMA, Qwen, OLMo) from 0.5B to 8B parameters, and two vision encoders (CLIP, SigLIP), PORTAL matches the standard pipeline trained on 558k image-caption pairs and exceeds it in three out of seven settings, peaking at 106% of the trained baseline. When PORTAL is used to initialize rather than replace projector alignment, the performance of the baseline is reached with $135\times$ fewer alignment FLOPs. Performance gaps are concentrated on OCR-heavy benchmarks, where marginal statistics appear insufficient to capture fine-grained character-level spatial structure that paired supervision provides.

2 Related Work

Projector alignment in multimodal LLMs. Two-stage multimodal LLM training, where a vision encoder is first aligned to a frozen language model via training a projector, followed by visual instruction tuning, has become a standard pipeline for training multimodal LLMs. This pipeline was popularized by LLaVA [1, 2] and replicated in InstructBLIP [9], BLIP-2 [10], Flamingo [11], Molmo [3], InternVL [4], and the Qwen-VL series [5]. Despite its success, the role of Stage-I projector alignment has been poorly understood and prior studies offer conflicting views. Prismatic finds it can be skipped without a loss [6]. Cambrian-1 reports that scaling Stage-I data to 2.5M pairs helps [7]. MM1 emphasizes resolution and data mixing over the projector itself [8]. We resolve this disagreement by identifying what projector alignment actually does: Stage-I acts as *geometric conditioning of visual features into the LLM input distribution*, not as fine-grained semantic learning. This explains the three findings above as a single phenomenon: skipping Stage-I works when the projector already lands in a compatible region (Prismatic [6]), more data helps only until that region is reached (Cambrian-1 [7]), and the resolution and data mixing dominate because they affect the conditioning more than the projector capacity does (MM1 [8]). These findings lead us to propose a closed-form projector that requires neither paired data nor gradient updates.

Connector designs and projector geometry. Beyond the MLP used in LLaVA [1], connector architectures span the Q-Former [10], Perceiver Resampler [11], locality-enhanced abstractors [12], ALIGN [13], vision-expert mixtures [14], prefix-attention adapters [15], and pixel-shuffle or patch-merging schemes for token reduction [16, 5]. A complementary line of work analyzes what these

connectors actually do. Verma et al. [17] track how visual information degrades through LLM layers, Li et al. [18] probe information loss through learned projectors, Li et al. [19] report systematic norm discrepancies between visual and text tokens, Shukor and Cord [20] show that cross-modal alignment emerges primarily after self-attention layers, and Krojer et al. [21] find that projected visual tokens have interpretable nearest neighbors in the LLM's embedding and intermediate-layer spaces. We use the 2-layer MLP projector of LLaVA by default, and also test PORTAL with a linear projector and with the C-Abstractor [12], which has $19\times$ more parameters (Section 5.3).

Optimal transport for feature alignment. The Bures-Wasserstein optimal transport map between two Gaussians has a closed form on positive-definite covariances [22, 23], which the construction of the projector in our work exploits. Courty et al. [24] introduced OT for domain adaptation, and Alvarez-Melis and Jaakkola [25] used Gromov-Wasserstein to align word-embedding spaces with no paired anchors. Closest to PORTAL is CORAL [26, 27], which aligns second-order statistics with a single linear map for domain adaptation. PORTAL extends this idea to the multimodal-alignment setting and ties it explicitly to the Bures-Wasserstein optimum on a shared top-k principal subspace, which we then use as a closed-form projector rather than a regularizer.

Training-free and efficient connector construction. Several recent works try to eliminate or reduce connector training. VPGTrans [28] transfers a learned visual prompt generator across LLMs at reduced cost. LangBridge [29] interprets visual tokens as combinations of language embeddings, enabling cross-LLM transfer at the price of bottlenecking visual information through a vocabulary subspace. SEMI [30] meta-learns a hypernetwork connector that adapts to new modalities from a few samples. In the related CLIP-classification setting, training-free or lightweight adapters such as Tip-Adapter [31] and TaskRes [32] cache or residualize over a frozen backbone. PORTAL differs from all of the above by requiring neither paired data, gradient steps, learned caches, nor a vocabulary bottleneck: it is a closed-form construction from per-modality summary statistics alone. Gupta et al. [33] also learn without paired multimodal data, but to improve a unimodal representation by sharing parameters across unpaired modalities, whereas we ask whether the projector between a frozen vision encoder and a frozen LLM needs paired training at all.

3 What Does the Projector Alignment *Actually* Do?

3.1 Projector Alignment Saturates with a Small Number of Samples

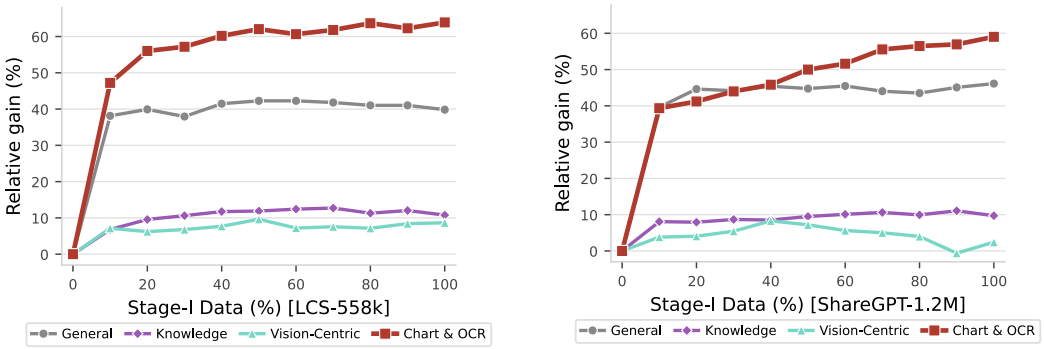
To investigate the role of projector alignment, we first examine how downstream performance scales with the amount of Stage-I training data. If strong performance requires large-scale data, it would suggest the projector is learning fine-grained visual-language correspondences. Conversely, if a small fraction of the data is sufficient, this would indicate that the projector primarily learns a coarse-grained correspondence, placing visual features into a broadly compatible input space for the language model.

We conduct this analysis on the LLaVA 1.5 setup (CLIP ViT-L/14-336, Vicuna 1.5 7B, 2-layer MLP projector), with LCS-558k for Stage-I and LLaVA-665k for Stage-II. For the data-fraction experiment we replace Vicuna with MobileLLaMA 1.4B, which keeps 11 data fractions on two corpora tractable. We evaluate on 16 benchmarks grouped into four categories adapted from the Cambrian-1 [7] taxonomy: Knowledge (Know.), General (Gen.), OCR, and Vision-Centric (Vis.). Appendix C lists the benchmarks in each. Full experimental details are in Appendix C.1.

We observe that only 10 to 20% of the LCS-558k image-caption pairs already capture most of the downstream benefit, and all categories plateau by 30 to 40% (Figure 2). The marginal downstream benefit of the remaining pairs is small under this protocol. ShareGPT-1.2M, more than twice as large, does not do better: its average peaks at 38.1 with 40% of its data (Appendix D). These results suggest that projector alignment is not primarily learning fine-grained visual-language correspondences. Rather, it appears to perform a coarse-grained alignment step, mapping visual features into a region of the LLM input space from which subsequent joint training can effectively adapt the full multimodal model.

3.2 Projector Alignment Admits a Broad Solution Space

We have shown that projector alignment can be achieved with only a small fraction of the standard pretraining data. This observation is consistent with two possible explanations: either the set of effective projector solutions is narrow but easy to reach, or the solution space itself is broad. To



(a) LCS-558k (b) ShareGPT-1.2M

Figure 2: **Stage-I gain saturates early.** Relative gain in downstream performance over the no-pretraining baseline by Cambrian-1 category, as a function of Stage-I data fraction on LCS-558k (left) and ShareGPT-1.2M (right). OCR and Chart benefit the most, while Knowledge and Vision-Centric scores saturate early.

distinguish between these possibilities, we examine whether independently trained projectors can be linearly interpolated while preserving performance.

Specifically, we train two projectors on different datasets (LCS-558k versus ShareGPT-1.2M), obtaining the weights W_{LCS} and W_{ShareGPT}. Then we evaluate the linear interpolation $W_\alpha = (1-\alpha)W_{\text{LCS}} + \alpha W_{\text{ShareGPT}}$ with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. If effective projectors occupied a narrow and isolated region, interpolation would be expected to degrade performance. Instead, we find that projectors along the entire interpolation path largely preserve downstream performance, supporting the view that projector alignment admits a broad solution space (Figure 3).

	Know.	Gen.	OCR	Vis.	All
LCS ($\alpha = 0$)	45.7	68.0	27.1	51.3	48.0
$\alpha = 0.25$	**46.2**	**68.5**	27.6	52.5	48.7
$\alpha = 0.50$	**46.2**	68.4	28.1	53.0	**49.0**
$\alpha = 0.75$	45.6	67.5	**28.4**	**54.0**	48.9
SGPT ($\alpha = 1$)	45.4	67.9	28.2	53.8	48.8

Figure 3: **Cross-data projectors are linearly interpolable.** Cross-data interpolation $W_\alpha = (1-\alpha)W_{\text{LCS}} + \alpha W_{\text{ShareGPT}}$, evaluated per Cambrian-1 category. The optimal α varies by category, but *every* interior blend outperforms both endpoints on the aggregate score, with the α=0.5 mixture peaking above either trained projector.

The per-category breakdown results further support this interpretation. The cross-data interpolation path is not only traversable but often outperforms either endpoint. Averaged across the four Cambrian-1 categories, every interior α improves on both LCS-558k and ShareGPT-1.2M projectors, with the optimum at α=0.5. The category-wise optima differ: OCR peaks closer to the ShareGPT endpoint, Knowledge closer to the LCS endpoint. The aggregate score improves with mixing, so a simple combination of two independently trained solutions, with no additional Stage-I optimization, outperforms both. Such behavior indicates that projector alignment admits a broad solution space.

The two projectors above start from the same initialization and differ only in their training data. We next asked whether the initialization matters in the same way, and trained a second pair on the same LCS-558k data from two different random initializations. Unlike the cross-data pair, this cross-seed pair is not linearly interpolable: captioning CIDEr drops from about 80 at both endpoints to 0.7 at α=0.5 (Appendix C.2). Initialization thus appears to select the loss basin, and training data the solution within it.

These results clarify the role of projector alignment. The early saturation with limited data and the traversable interpolation path between projectors trained on different data suggest that projector alignment does not require a highly specific solution. Instead, projector alignment appears to place visual features into a broadly compatible region of the LLM input space, where many different projector weights can serve the same functional role.

4 PORTAL: Training-Free Projector via Optimal Transport

The previous findings suggest that projector alignment primarily serves as geometric conditioning, placing visual features into a region of the LLM input space from which subsequent joint training can effectively adapt the multimodal model. This mapping admits a large set of solutions, motivating a simpler alternative to the projector alignment training stage.

We instantiate this idea with PORTAL (**P**rojector via **O**ptimal-transpo**RT AL**ignment), a training-free method that initializes the projector from the marginal feature statistics of the vision encoder and the LLM. PORTAL estimates the mean, principal directions, and variances of each modality separately, then constructs a closed-form optimal-transport map that aligns the source visual feature distribution with a compatible target language-feature distribution. We also show why matching only means and covariances can suffice. The LLM normalizes each token before attention, so attention depends only on token directions. When token norms vary little, the spread of these directions is fixed by the covariance, which PORTAL matches.

4.1 Closed-Form Projector Initialization

Let $\mathbf{x} \in \mathbb{R}^{d_s}$ denote a feature from the source modality (*e.g.*, a per-patch vision token in our setting) and $\mathbf{y} \in \mathbb{R}^{d_t}$ a target embedding (*e.g.*, an LLM input token). We seek a map $f : \mathbb{R}^{d_s} \to \mathbb{R}^{d_t}$, the *projector* in standard multimodal-LLM terminology, whose output distribution is compatible with the distribution the target is trained to process. We adopt the standard 2-layer MLP used by LLaVA [2],

$$f(\mathbf{x}) = \mathbf{W}_2\, \phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2, \quad \mathbf{W}_1 \in \mathbb{R}^{d_t \times d_s},\ \mathbf{W}_2 \in \mathbb{R}^{d_t \times d_t}, \tag{1}$$

with ϕ as GELU. We will fix $\mathbf{W}_2 = \mathbf{I}_{d_t}$ (*i.e.*, the identity matrix of dimension d_t), $\mathbf{b}_2 = \mathbf{0}$ (*i.e.*, the zero vector of dimension d_t), and construct $(\mathbf{W}_1, \mathbf{b}_1)$ in closed form.

PORTAL. Approximate each modality's marginal as a Gaussian, $\mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ on the source side and $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ on the target side. We derive the affine parameters $(\mathbf{W}_1, \mathbf{b}_1)$ from the Gaussian optimal transport map [34, 22] between these approximations, computed on the top-k PCA subspaces of source and target with coordinates identified by eigenvalue order:

$$\mathbf{W}_1 = \mathbf{V}_t\, \mathrm{diag}\left(\frac{\boldsymbol{\sigma}_t}{\boldsymbol{\sigma}_s}\right) \mathbf{V}_s^\top, \qquad \mathbf{b}_1 = \boldsymbol{\mu}_t - \mathbf{W}_1\boldsymbol{\mu}_s, \tag{2}$$

where $\mathbf{V}_s \in \mathbb{R}^{d_s \times k}$ and $\mathbf{V}_t \in \mathbb{R}^{d_t \times k}$ are the top-k PCA bases of the source and target features (orthonormal columns) and $\boldsymbol{\sigma}_s, \boldsymbol{\sigma}_t \in \mathbb{R}^k$ are the square roots of the top-k covariance eigenvalues, *i.e.*, the spread along each principal axis. The map $\mathbf{W}_1$ whitens the source distribution (decorrelates it and rescales each axis to unit variance), expresses it in the target's principal axes, and rescales each axis to match the target's spread. The bias $\mathbf{b}_1$ recenters the result. The closed form matches Gaussian moments at the preactivation $\mathbf{W}_1\mathbf{x} + \mathbf{b}_1$. The post-GELU output is verified empirically in Appendix H.4. Appendix F gives a self-contained PyTorch implementation.

This construction depends only on summary statistics of each modality (*i.e.*, means, principal directions, and the spread along each direction). It requires no Stage-I paired captions and no Stage-I optimization. A single PCA on each side is sufficient. It is agnostic to which model produces $\mathbf{x}$ and which model consumes $f(\mathbf{x})$. We adopt k=512 as the default throughout the paper, which is small enough to suppress noisy directions and large enough to preserve the dominant geometry. Varying k over $\{64, 128, 256, 512, 1024\}$ supports this choice empirically (Appendix K).

On the Gaussian assumption. Eq. (2) treats both modalities as Gaussian to obtain a closed-form transport map in terms of means, principal directions, and the spread along each direction. Real source and target distributions deviate from Gaussianity in tail weight and in the shape of their marginals (Appendix H.1). We explain why the construction remains effective under these deviations in Section 4.2.

Targeting hidden states instead of input embeddings. Eq. (2) matches visual features to the LLM's input embeddings, the vectors the LLM looks up for each text token before its first transformer block. The same construction can instead target the hidden states that text tokens have after block ℓ: we replace $(\boldsymbol{\mu}_t, \mathbf{V}_t, \boldsymbol{\sigma}_t)$ by the mean, PCA basis, and spread along the principal axes of these hidden states, collected by passing a list of English words and short phrases through the LLM. The visual tokens still enter the LLM at its input. Only the target statistics change. Trained projectors that inject

visual tokens into deeper layers [35, 36] motivate this variant, which we call PORTAL-ctxℓ. We use it to ablate the target layer (Section 5.4), where the input embeddings (ℓ=0) remain the best target. Details are in Appendix G.

4.2 Why Moment Matching Survives Non-Gaussianity

Real vision and text features are not Gaussian (Appendix H.1), so Eq. (2) cannot match their full distributions. We argue that it does not need to. Every LLM we use normalizes before computing attention logits (RMSNorm on each input token, or on queries and keys in OLMo 2), so rescaling a token leaves its logits unchanged. Attention therefore depends only on the directions of the tokens, and the relevant property of the projector output is how these directions are spread.

We measure this spread by the *cosine dispersion* ω, the standard deviation of the cosine similarity between two tokens:

$$\omega^2 = \frac{1}{\binom{n}{2}} \sum_{i<j} \left(\cos\angle(\mathbf{z}_i, \mathbf{z}_j) - \overline{\cos}\right)^2, \qquad \overline{\cos} = \frac{1}{\binom{n}{2}} \sum_{i<j} \cos\angle(\mathbf{z}_i, \mathbf{z}_j), \tag{3}$$

where $\{\mathbf{z}_i\}_{i=1}^n$ are the output tokens, pooled over all patches of all images. A small ω means the tokens point in similar directions, a large ω that they spread over many directions. Two properties of a distribution with covariance $\mathbf{\Sigma}$ determine ω. The *effective dimension* $d_{\text{eff}}(\mathbf{\Sigma}) := (\operatorname{tr}\mathbf{\Sigma})^2/\operatorname{tr}(\mathbf{\Sigma}^2)$ counts the directions that carry most of the variance: it equals d when variance is spread evenly over d directions and 1 when a single direction holds it all. The *norm spread* $\kappa := \operatorname{std}(\|\mathbf{z}\|^2)/\mathbb{E}[\|\mathbf{z}\|^2]$ is small when all tokens have about the same length.

Proposition 1 (Spectrum-to-ω bound)**.** *Let* $\mathbf{z}$ *and* $\mathbf{z}'$ *be independent random vectors in* $\mathbb{R}^d$ *with the same distribution, which has mean zero and covariance* $\mathbf{\Sigma}$*. Assume* $\mathbb{P}(\mathbf{z} = \mathbf{0}) = 0$*. Then*

$$\left|\operatorname{Var}\big(\cos\angle(\mathbf{z}, \mathbf{z}')\big) - \frac{1}{d_{\text{eff}}(\mathbf{\Sigma})}\right| \le 2\kappa + \kappa^2 + \kappa^4. \tag{4}$$

The assumption $\mathbb{P}(\mathbf{z} = \mathbf{0}) = 0$ says that a token is never exactly the zero vector, so that its cosine with another token is always defined. The left side of Eq. (4) is the variance of the cosine similarity between two tokens drawn at random from the distribution, which is the population value of ω^2 that Eq. (3) estimates from n tokens. The right side is zero when all tokens have the same length ($\kappa = 0$), and the proposition then gives $\omega^2 = 1/d_{\text{eff}}(\mathbf{\Sigma})$: the spread of token directions is fixed by the covariance alone. Intuitively, when all tokens have the same length, the cosine of two tokens is their dot product divided by a constant, and the second moment of the dot product of two independent tokens depends only on the covariance. The proof is in Appendix H.2 and does not assume Gaussianity. PORTAL matches the target's mean and covariance on the top-k subspace (second-order matching, *i.e.*, matching the means and covariances of the two distributions), so in this limit its outputs inherit the target's ω.

Real token norms vary (Appendix H.4), so the bound does not predict ω precisely for our features. We therefore measure ω directly. On three (CLIP, LLM) pairs, PORTAL reduces ω from 0.23 at the source to 0.09 to 0.10, towards the target's 0.05 to 0.06, while a random projector leaves ω at the source value (Appendix H.3). The attention weights also depend on the learned query and key projections, which ω does not describe.

5 Experiments

We test whether PORTAL, built without any paired image-caption data, matches projector alignment trained on 558k image-caption pairs across LLM backbones and vision encoders.

We evaluate this question by inserting a PORTAL-constructed projector into the standard LLaVA pipeline and then running the same LLaVA-665k instruction tuning for every configuration. We compare against two baselines: *(i)* the standard trained LLaVA projector, obtained from 558k paired captions, and *(ii)* a randomly initialized projector which skips projector alignment entirely. This setup isolates the contribution of projector initialization: after initialization, all models receive identical joint training.

We test seven (vision encoder, LLM) pairs, built from three 7B to 8B LLMs (Vicuna 1.5 7B [37], LLaMA 3.1 8B [38], Qwen 2.5 7B [39]), three smaller LLMs for cross-scale checks (LLaMA 3.2 1B,

Table 1: **PORTAL generalizes across LLM backbones and vision encoders.** All configurations use identical Stage-II instruction tuning (LLaVA-665k [2]). PORTAL requires zero Stage-I paired captions and zero Stage-I gradient steps. Baseline †: uses 558k paired captions for Stage-I pretraining. **Bold**: best among zero training data methods. **Avg**: unweighted mean of the 16 per-benchmark scores. %BL: this average as a percentage of the trained baseline (BL), which is therefore 100%. Subscripts on the Avg and %BL columns give the 95% bootstrap confidence-interval half-width over per-sample outcomes, from 1000 resamples paired across methods. LLM backbones: Vicuna 1.5 7B [37], LLaMA 3.1 8B [38], Qwen 2.5 7B [39], LLaMA 3.2 1B [38], Qwen 2.5 0.5B [39], OLMo 2 1B [40]. Vision encoders: CLIP ViT-L/14-336 [41], SigLIP SO400M [42]. ^{L}: 500-sample Lite subset from `lmms-eval` [43].

				Knowledge				General				OCR				Vision					
LLM	**Scale**	**Vision**	**Init**	*AI2D*	*SciQA*	*MathV*	*MMMU*	*GQA*	*MME*	*SEED^{L}*	*MMB*	*TxtVQA*	*OCR-Bv2*	*DocVQA*	*ChartQA*	*CV-2D*	*CV-3D*	*BLINK*	*RealWQ*	**Avg**	**%BL**
Vicuna 1.5	7B	CLIP	Baseline†	54.5	67.9	26.1	34.3	66.6	74.5	67.7	63.2	57.3	12.5	22.0	16.5	54.7	58.3	37.4	54.9	48.0	100.0
			Random	51.5	**69.1**	27.2	32.8	64.2	69.2	57.2	58.1	52.8	11.0	17.1	15.2	52.5	53.8	34.7	51.9	$44.9_{\pm 0.60}$	$93.5_{\pm 1.6}$
			PORTAL	**53.4**	68.2	**27.4**	**34.4**	**66.4**	**69.9**	**67.9**	**60.4**	**55.1**	**12.7**	**20.5**	**16.3**	**56.2**	**60.7**	**36.0**	**53.5**	$\mathbf{47.4}_{\pm 0.56}$	$\mathbf{98.8}_{\pm 1.5}$
		SigLIP	Baseline†	55.8	68.0	27.2	34.0	68.7	73.6	69.1	66.0	61.5	13.3	25.3	19.1	57.5	60.8	28.1	56.6	49.0	100.0
			Random	51.8	67.9	26.7	**35.1**	56.8	60.6	46.5	52.3	43.8	7.9	8.9	12.8	47.1	57.5	25.6	45.9	$40.5_{\pm 0.58}$	$82.5_{\pm 1.1}$
			PORTAL	**55.5**	**68.3**	**27.9**	34.4	**67.8**	**71.6**	**68.3**	**64.3**	**60.0**	**13.6**	**24.8**	**17.9**	**57.6**	**64.6**	**28.6**	**54.2**	$\mathbf{48.7}_{\pm 0.56}$	$\mathbf{99.3}_{\pm 1.5}$
LLaMA 3.1	8B	CLIP	Baseline†	58.5	73.8	31.3	36.8	67.0	73.0	71.7	68.1	57.2	13.7	21.8	15.1	64.5	63.0	39.8	57.5	50.8	100.0
			Random	58.6	74.8	**31.7**	35.3	67.6	73.4	69.5	67.5	56.2	13.9	19.6	15.0	61.8	66.2	38.5	**57.0**	$50.4_{\pm 0.58}$	$99.2_{\pm 1.4}$
			PORTAL	**62.2**	**76.2**	30.7	**38.6**	**67.7**	**74.1**	**69.9**	**70.1**	**58.0**	**14.5**	**22.7**	**16.8**	**65.2**	**70.9**	**40.4**	56.9	$\mathbf{52.2}_{\pm 0.57}$	$\mathbf{102.7}_{\pm 1.5}$
LLaMA 3.2	1B	CLIP	Baseline†	46.5	66.2	26.9	33.6	59.7	58.4	56.2	46.1	43.7	10.1	12.8	10.8	53.5	56.4	37.2	47.3	41.6	100.0
			Random	46.6	66.9	26.3	**34.2**	59.8	58.1	56.0	45.2	43.8	10.4	12.8	11.0	54.2	54.9	37.4	46.9	$41.5_{\pm 0.57}$	$99.9_{\pm 0.7}$
			PORTAL	**47.6**	**67.1**	**29.2**	33.7	**62.1**	**66.0**	**59.6**	**54.2**	**47.1**	**11.4**	**16.3**	**12.0**	**54.6**	**57.3**	**38.9**	**49.7**	$\mathbf{44.2}_{\pm 0.60}$	$\mathbf{106.2}_{\pm 1.3}$
Qwen 2.5	7B	CLIP	Baseline†	66.9	77.5	33.3	44.7	65.6	74.6	70.7	71.2	57.2	15.6	23.0	19.0	54.0	69.0	42.3	56.6	52.6	100.0
			Random	62.1	73.9	27.7	42.0	63.7	70.9	67.7	60.1	51.0	13.6	17.6	14.6	56.3	60.0	**42.1**	53.9	$48.6_{\pm 0.56}$	$92.4_{\pm 0.9}$
			PORTAL	**65.7**	**76.7**	**31.9**	**44.8**	**65.5**	**75.9**	**70.9**	**68.0**	**55.4**	**14.9**	**18.9**	**17.1**	**57.9**	**64.8**	41.5	**56.1**	$\mathbf{51.7}_{\pm 0.58}$	$\mathbf{98.3}_{\pm 1.4}$
Qwen 2.5	0.5B	CLIP	Baseline†	43.6	58.6	26.7	31.7	52.2	53.9	47.9	38.7	35.7	11.1	9.4	9.7	45.7	50.7	39.4	46.0	37.6	100.0
			Random	43.5	58.5	26.8	**33.1**	52.5	54.2	48.9	37.9	35.6	10.5	9.5	9.8	**44.2**	**51.5**	**40.5**	45.5	$37.7_{\pm 0.58}$	$100.2_{\pm 1.0}$
			PORTAL	**44.2**	**59.1**	**27.4**	31.3	**58.1**	**57.6**	**56.0**	**46.4**	**41.8**	**11.8**	**14.6**	**11.6**	41.4	51.2	39.7	**45.6**	$\mathbf{39.9}_{\pm 0.56}$	$\mathbf{106.1}_{\pm 1.3}$
OLMo 2	1B	CLIP	Baseline†	46.9	64.3	25.0	33.9	62.7	63.1	63.2	57.6	46.3	10.5	17.2	12.2	57.3	54.8	37.6	48.1	43.8	100.0
			Random	45.7	58.8	24.6	31.2	59.4	59.5	59.8	49.4	42.4	9.7	13.6	11.0	50.2	50.9	**37.5**	43.9	$40.5_{\pm 0.57}$	$92.8_{\pm 1.0}$
			PORTAL	**47.2**	**61.8**	**26.8**	31.2	**61.8**	**61.8**	**60.2**	**51.5**	**44.2**	**9.8**	**15.0**	**11.1**	**54.8**	**55.3**	36.2	**46.3**	$\mathbf{42.2}_{\pm 0.59}$	$\mathbf{96.3}_{\pm 1.0}$

Qwen 2.5 0.5B, OLMo 2 1B [40]), and two vision encoders (CLIP ViT-L/14-336 [41], SigLIP-SO400M [42]). Table 1 lists the pairs. The trained baseline uses the standard LLaVA Stage-I corpus of 558k LCS image-caption pairs, and all configurations share the same LLaVA-665k instruction-tuning data for Stage-II. The 16 evaluation benchmarks are grouped into the four Cambrian-1 categories. The full benchmark list, citations, and evaluation protocol are in Appendix C.

5.1 PORTAL Matches Trained Projector Alignment Without Paired Data

We find that our closed-form projector PORTAL requiring *no paired captions or gradient updates* achieves comparable downstream performance to conventional alignment pipelines trained on 558k image-caption pairs, across a range of vision encoder and LLM combinations (Table 1). In particular, using LLaMA 3.2 1B/CLIP and Qwen 2.5 0.5B/CLIP, PORTAL reaches $\mathbf{106.2}\%$ and $\mathbf{106.1}\%$ of the trained baseline, respectively. Using LLaMA 3.1 8B/CLIP, it exceeds the trained baseline, reaching $\mathbf{102.7}\%$. On Vicuna 1.5 with CLIP and SigLIP, Qwen 2.5 7B, and OLMo 2 1B, PORTAL remains very close to the trained projector, reaching 98.8%, 99.3%, 98.3%, and 96.3% of the trained baseline, respectively. Thus, PORTAL removes the requirement for paired training data and optimization cost of projector alignment while preserving almost all of its downstream benefit after instruction tuning.

The advantage of PORTAL is clearest where the random projector fails. On SigLIP, the Random baseline collapses to $\mathbf{82.5}\%$ of the trained projector. PORTAL closes that gap to $\mathbf{99.3}\%$. This is what we would expect if the alignment stage were dominated by distributional matching rather than caption supervision.

Two settings sit below the trained baseline's bootstrap CI, Qwen 2.5 7B/CLIP and OLMo 2 1B/CLIP, with the gaps concentrated on OCR and Chart benchmarks. This matches the observation in Section 3.1 that OCR and Chart drive most of the Stage-I gain over the no-pretraining baseline, supporting our reading that character-level spatial structure is the regime where paired captions teach what marginal statistics alone cannot recover. Importantly, this phenomenon is not specific to PORTAL. Other unpaired distribution-matching approaches, including CORAL [27] and Gromov-Wasserstein [25] alignment, also recover most of the performance of paired Stage-I training (Appendix E.1). This

provides additional evidence that much of the benefit of Stage-I can be obtained by bringing the two modalities into a compatible feature distribution, rather than requiring a uniquely learned image-text correspondence.

A category-level analysis across all seven configurations further shows that this behavior is not systematic across OCR tasks: PORTAL retains, on average, 100.7%, 102.0%, 102.4%, and 100.4% of the trained-baseline performance on Knowledge, General, OCR, and Vision-Centric tasks, respectively (Appendix E.2). We further test PORTAL beyond the standard LLaVA-665k Stage-II setting. Under substantially larger-scale multimodal instruction tuning with 3.2M samples, PORTAL slightly outperforms the conventionally Stage-I-trained initialization (Appendix E.3). We also evaluate object hallucination on POPE [44] and find that PORTAL matches or exceeds the trained Stage-I baseline in every setting (Appendix E.4).

5.2 PORTAL Reduces the Projector Alignment Budget

The previous experiment shows that PORTAL can provide a strong projector without paired optimization. We next ask whether PORTAL can also serve as a warm start for projector alignment. Instead of using PORTAL to skip this stage entirely, we initialize the projector with PORTAL and then continue projector alignment with progressively smaller subsets of LCS-558k. This experiment evaluates whether closed-form alignment can reduce the amount of paired data needed for subsequent projector training.

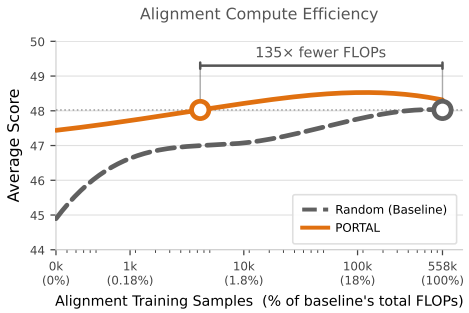


Figure 4: **Alignment compute efficiency** on Vicuna 1.5 7B/CLIP. Average over the 16 benchmarks as a function of the projector alignment budget, with identical Stage-II tuning on LLaVA-665k. PORTAL stays above Random at every budget. The marked point indicates the crossover at roughly 4k captions, where PORTAL matches the 558k-trained baseline at $\sim 135\times$ fewer alignment FLOPs. The horizontal dashed line shows the 558k baseline.

We vary the projector alignment budget over $\{0, 1\text{k}, 10\text{k}, 100\text{k}, 558\text{k}\}$ paired captions from LCS-558k on Vicuna 1.5 7B/CLIP, starting from either a Random or a PORTAL projector. The 0 point skips projector alignment entirely and proceeds directly to Stage-II from the initialization, and the 558k point is the standard LLaVA baseline. Stage-II tuning on LLaVA-665k is identical for every configuration.

PORTAL provides a better starting point at every projector alignment budget than the random initialization (Figure 4). With zero projector training, it already outperforms random initialization. As more paired captions are added, the PORTAL-initialized projector remains consistently above the randomly initialized projector. PORTAL reaches the performance of the full 558k trained projector baseline using only around 4k captions, corresponding to roughly $\mathbf{135}\times$ fewer projector alignment FLOPs. Beyond matching the trained baseline at lower compute, PORTAL with as few as 10k paired captions exceeds the full 558k-trained projector baseline and continues to widen the gap up to 100k. PORTAL therefore offers two practical advantages: it reaches the trained baseline at substantially lower alignment cost, and it surpasses that baseline once even a small paired-data budget is available.

This result strengthens the main claim that much of projector alignment is a distributional conditioning problem. The projector has a large solution space, and many mappings can make visual features compatible with the LLM input space. PORTAL identifies one such compatible solution directly in closed form. This solution is not necessarily the exact optimum selected by supervised projector training but it lies close to the high-performing region, providing strong performance without paired optimization and a small amount of paired data can further refine it within this region. Additional projector training thus mainly refines a projector that is already compatible.

5.3 Ablations: Projector Architecture, Image Pool, and Rank

A key advantage of PORTAL is that it is a closed-form geometric construction rather than a learned projector tied to a particular architecture or paired dataset. We therefore test whether its gains persist when the projector is simplified and when the unlabeled image pool used to estimate vision-side statistics is changed.

Table 2: **Linear vs. MLP-2x-GELU projector** on LLaMA 3.2 1B/CLIP. Full per-benchmark breakdown in Appendix J.

Projector	Init	Avg	%BL
MLP	Trained†	41.6	100.0
	Random	41.5	99.9
	PORTAL	**44.2**	**106.2**
Linear	Trained†	39.9	100.0
	Random	37.6	94.2
	PORTAL	**44.7**	**111.8**

For architecture, we replace the standard 2-layer MLP-GELU projector with a single linear layer, keeping all other components fixed (Table 2). If PORTAL's gains depended on the expressivity of the MLP, they should weaken in this setting. Instead, the gain becomes larger. With the MLP projector, PORTAL reaches $\mathbf{106.2}\%$ of the trained baseline against Random's 99.9%. With the linear projector, it reaches $\mathbf{111.8}\%$ against Random's 94.2%, widening the Random-to-PORTAL gap from 6.3 to 17.6 percentage points. This shows that PORTAL's advantage does not come from a particular nonlinearity or hidden expansion. It comes from placing the visual features into a better-matched LLM input distribution. We also test the opposite regime by substantially increasing connector capacity. Using the C-Abstractor [12], which contains approximately $19\times$ more parameters than the two-layer MLP and additionally performs token mixing and sequence compression, PORTAL remains essentially indistinguishable from conventional Stage-I training (42.93 vs. 42.98 average score; Appendix E.5).

For the image pool, we vary the unlabeled image pool used to estimate the vision-side statistics. On LLaMA 3.2 1B/CLIP, we compare a small single-domain pool, COCO `val2017` with 1024 images, against a larger and more diverse Cambrian-737k subset spanning 9 sources with 4096 images. The small COCO pool matches or exceeds the diversified pool on 12 of 15 benchmarks and obtains a higher average score ($\mathbf{44.0}$ vs. 43.4). This suggests that PORTAL does not require a carefully curated or highly diverse image pool. A small unlabeled natural-image pool is sufficient to estimate the source statistics well enough for downstream instruction tuning in this setting. Per-benchmark results are reported in Appendix I.

Additionally, the PORTAL construction depends on a rank k used to define the PCA subspace. We vary k over $\{64, 128, 256, 512, 1024\}$ on LLaMA 3.2 1B/CLIP, constructing each projector from the same 1024 COCO images and running identical Stage-II tuning. The method is robust to this choice (Appendix K): every k outperforms the trained baseline by at least $\sim 4\%$. Performance is flat from k=64 to k=256 ($\sim$43.3 Avg), peaks at k=512 (44.2), and decreases marginally at k=1024 (44.0). The plateau at low ranks indicates that the dominant PCA directions already capture most of the cross-modal structure. The small drop at k=1024 is consistent with noisy eigenvectors diluting the transport map. We adopt k=512 as the default.

5.4 Target Layer Choice

PORTAL targets the LLM's input embedding distribution by default. The PORTAL-ctxℓ variant (Section 4) generalizes the construction to deeper targets, matching visual features to hidden states after LLM block ℓ instead. These hidden states are collected by passing the concept corpus of Krojer et al. [21] through the LLM (Appendix G). We use this variant to ablate the choice of target layer.

Table 3: **Layer ablation for PORTAL-ctx** on LLaMA 3.1 8B/CLIP. Avg over the 16-benchmark suite, %BL relative to trained Stage-I.

Target	Avg	%BL
Trained†	51.3	100.0
Random	51.4	100.2
ℓ=0 (embed)	**52.5**	**102.3**
ℓ=4	52.3	101.9
ℓ=8	52.0	101.4

Targeting the embedding table ($\ell = 0$) is the strongest choice on LLaMA 3.1 8B/CLIP, reaching $\mathbf{102.3}\%$ of the trained projector alignment baseline (Table 3). Contextualized targets at $\ell = 4$ and $\ell = 8$ remain close, reaching 101.9% and 101.4%, respectively. We therefore use the embedding table as the default target.

6 Conclusion

This work rethinks projector alignment in multimodal LLMs. Our results suggest that this stage primarily performs coarse alignment, mapping visual features into a language-model-compatible input space rather than learning fine-grained visual-language correspondences. This view is supported by the early saturation of alignment data, where 10 to 20% of LCS-558k recovers most downstream gains, and by the linear interpolability of projectors trained on different datasets from a shared initialization.

Motivated by this finding, we introduced PORTAL, a training-free projector initialization. PORTAL uses only per-modality summary statistics, requires no paired image-caption data or gradient-based optimization, and remains close to the 558k-trained baseline across the evaluated model pairs. As an initialization, it reaches the trained baseline with roughly $135\times$ lower alignment compute. These results suggest that projector alignment can often be replaced by, or reduced to, a lightweight feature-distribution matching step, lowering the cost and paired-data requirements for building multimodal LLMs.

Several natural directions remain for future work. Our study focuses on projector alignment with frozen vision and language backbones during alignment, single-image inputs, and simple projector architectures. Extending the analysis to richer connectors, long-context grounding, multi-image reasoning, and additional modalities such as audio, video, and 3D would further clarify how broadly this coarse-alignment view applies. We discuss other limitations of PORTAL in Appendix A.

Acknowledgements

We thank Yulun Jiang, Siba Smarak Panigrahi, and Petar Damjanović for their valuable suggestions, which helped improve the clarity of the manuscript. We gratefully acknowledge the support of the Swiss National Science Foundation (SNSF) starting grant TMSGI2_226252/1, SNSF grant IC00I0_231922, SNSF grant 10.004.411, Swiss Cancer League (KFS-6340-02-2025-R), Kamprad Foundation (Cancer Immunotherapy Networking Program), and the Swiss AI Initiative Large Call (ab006). Maria Brbić is a CIFAR Fellow in the Multiscale Human Program.

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[2] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[3] Matt Deitke et al. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Vision-Language Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[4] Zhe Chen et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[6] Siddharth Karamcheti et al. Prismatic VLMs: Investigating the Design Space of Visually-Conditioned Language Models. In *International Conference on Machine Learning (ICML)*, 2024.

[7] Shengbang Tong et al. Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[8] Brandon McKinzie et al. MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training. In *European Conference on Computer Vision (ECCV)*, 2024.

[9] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *International Conference on Machine Learning (ICML)*, 2023.

[11] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: A Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[12] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-Enhanced Projector for Multimodal LLM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] Ahmed Masry, Juan A Rodriguez, Tianyu Zhang, Suyuchen Wang, Chao Wang, Aarash Feizi, Akshay Kalkunte Suresh, Abhay Puri, Xiangru Jian, Pierre-André Noël, et al. AlignVLM: Bridging Vision and Language Latent Spaces for Multimodal Document Understanding. *arXiv preprint arXiv:2502.01341*, 2025.

[14] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. MoVA: Adapting Mixture of Vision Experts to Multimodal Context. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[15] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. In *International Conference on Learning Representations (ICLR)*, 2024.

[16] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. SmolVLM: Redefining Small and Efficient Multimodal Models. *arXiv preprint arXiv:2504.05299*, 2025.

[17] Gaurav Verma, Minje Choi, Kartik Sharma, Jamelle Watson-Daniels, Sejoon Oh, and Srijan Kumar. Cross-Modal Projection in Multimodal LLMs Doesn't Really Project Visual Attributes to Textual Space. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[18] Wenyan Li, Raphael Tang, Chengzu Li, Caiqi Zhang, Ivan Vulić, and Anders Søgaard. Lost in Embeddings: Information Loss in Vision-Language Models. In *Findings of the Association for Computational Linguistics (EMNLP)*, 2025.

[19] Bozhou Li et al. The Unseen Bias: How Norm Discrepancy in Pre-Norm MLLMs Leads to Visual Information Loss. *arXiv preprint arXiv:2512.08374*, 2025.

[20] Mustafa Shukor and Matthieu Cord. Implicit Multimodal Alignment: On the Generalization of Frozen LLMs to Multimodal Inputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[21] Benno Krojer, Shravan Nayak, Oscar Mañas, Vaibhav Adlakha, Desmond Elliott, Siva Reddy, and Marius Mosbach. LatentLens: Revealing Highly Interpretable Visual Tokens in LLMs. *arXiv preprint arXiv:2602.00462*, 2026.

[22] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures-Wasserstein Distance Between Positive Definite Matrices. *Expositiones Mathematicae*, 2019.

[23] Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 2019.

[24] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal Transport for Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[25] David Alvarez-Melis and Tommi S. Jaakkola. Gromov-Wasserstein Alignment of Word Embedding Spaces. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[26] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of Frustratingly Easy Domain Adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[27] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision (ECCV) Workshops*, 2016.

[28] Ao Zhang et al. VPGTrans: Transfer Visual Prompt Generator across LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[29] Jiaqi Liao, Yuwei Niu, Fanqing Meng, Hao Li, Changyao Tian, Yinuo Du, Yuwen Xiong, Dianqi Li, Xizhou Zhu, Li Yuan, Jifeng Dai, and Yu Cheng. LangBridge: Interpreting Image as a Combination of Language Embeddings. In *International Conference on Computer Vision (ICCV)*, 2025.

[30] Osman Batur İnce, André F. T. Martins, Oisin Mac Aodha, and Edoardo M. Ponti. Sample-Efficient Integration of New Modalities into Large Language Models. *arXiv preprint arXiv:2509.04606*, 2025.

[31] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-Adapter: Training-Free Adaption of CLIP for Few-shot Classification. In *European Conference on Computer Vision (ECCV)*, 2022.

[32] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task Residual for Tuning Vision-Language Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[33] Sharut Gupta, Shobhita Sundaram, Chenyu Wang, Stefanie Jegelka, and Phillip Isola. Better together: Leveraging unpaired multimodal data for stronger unimodal models. In *International Conference on Learning Representations (ICLR)*, 2026.

[34] Ingram Olkin and Friedrich Pukelsheim. The Distance Between Two Random Vectors with Given Dispersion Matrices. *Linear Algebra and its Applications*, 1982.

[35] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[36] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

[37] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, Ion Stoica, and Eric P Xing. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. *LMSYS Blog*, 2023.

[38] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

[39] An Yang, Baosong Yang, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

[40] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, et al. 2 OLMo 2 Furious. *arXiv preprint arXiv:2501.00656*, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[42] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. *arXiv preprint arXiv:2303.15343*, 2023.

[43] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. In *Findings of the Association for Computational Linguistics (NAACL)*, 2025.

[44] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[45] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. In *European Conference on Computer Vision (ECCV)*, 2024.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 2014.

[47] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A Diagram Is Worth a Dozen Images. In *European Conference on Computer Vision (ECCV)*, 2016.

[48] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[49] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

[50] 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. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[51] Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[52] 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. *arXiv preprint arXiv:2306.13394*, 2024.

[53] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking Multimodal Large Language Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[54] 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? In *European Conference on Computer Vision (ECCV)*, 2024.

[55] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[56] Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. *arXiv preprint arXiv:2501.00321*, 2025.

[57] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A Dataset for VQA on Document Images. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[58] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics (ACL)*, 2022.

[59] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal Large Language Models Can See but Not Perceive. In *European Conference on Computer Vision (ECCV)*, 2024.

[60] xAI. RealWorldQA, 2024.

[61] Bradley Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 1979.

[62] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and Chunhua Shen. MobileVLM: A Fast, Strong and Open Vision Language Assistant for Mobile Devices. *arXiv preprint arXiv:2312.16886*, 2023.

[63] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based Image Description Evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[64] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

[65] K. V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 1970.

[66] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 1952.

[67] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, 2007.

A Limitations

Our analysis and method address a specific slice of the multimodal-LLM pipeline: Stage-I projector alignment with frozen vision and language backbones. Several caveats follow from that scope. *(i)* The empirical analysis in Section 3 runs on the LLaVA 1.5 architecture with a 2-layer MLP projector. Richer connectors (resamplers, perceiver-style cross-attention, multi-stage abstractors) may not exhibit the same linear interpolability, and we do not test them here. *(ii)* PORTAL is constructed from a Gaussian approximation of each modality. The approximation captures means, principal directions, and per-axis spread, but no higher-order structure. Appendix H.1 quantifies how far real source and target distributions deviate from this assumption. We hypothesize that the residual gap on OCR and Chart benchmarks reflects this missing higher-order structure (character-level spatial layout). *(iii)* As in LLaVA-1.5, all experiments keep the vision encoder frozen and train only the projector and the LLM in Stage-II. We do not study end-to-end fine-tuning that also updates the vision encoder. *(iv)* Our 16-benchmark suite follows the Cambrian-1 grouping and inherits its biases. It does not stress long-context grounding or multi-image reasoning, which may behave differently. *(v)* We instantiate PORTAL for vision-to-language alignment. We have not tested it on audio or video encoders. The present results should be regarded as a first existence proof rather than a finished construction. *(vi)* Our reading of Stage-I as geometric conditioning is the simplest hypothesis consistent with our data. We do not claim it is the unique mechanism, and other interpretations remain possible. *(vii)* The closed-form construction exceeds trained Stage-I or stays within 1.6 average points of it across the configurations we test. We make no broader claim that this holds across all multimodal-LLM regimes. Richer benchmarks or backbones may reveal regimes where trained Stage-I genuinely outperforms PORTAL.

B Training Details and Compute Resources

All experiments follow the standard two-stage LLaVA-1.5 [2] training recipe. Training is performed on $8\times$ NVIDIA A100 80 GB GPUs with DeepSpeed ZeRO-2 and bf16 mixed precision.

Hyperparameters. Table 4 summarizes the training configuration for both stages. All runs use the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$ and no gradient clipping.

Table 4: **Training hyperparameters.** Identical to LLaVA-1.5.

	Stage I (Pretrain)	**Stage II (Finetune)**
Learning rate	1×10^{-3}	2×10^{-5}
Schedule	Cosine	Cosine
Warmup	200 steps	3% ratio
Weight decay	5×10^{-2}	0
Per-device batch size	32	4
Gradient accumulation	1	4
Effective batch size	256	128
Epochs	1	1
Trainable parameters	Projector only	Projector + LLM

Compute budget. Each Stage-I run completes in under one hour for MobileLLaMA 1.4B and in under four hours for Vicuna 7B on $8\times$A100 GPUs. Stage-II instruction tuning takes between two and ten hours depending on model scale (2.4 h for Qwen 0.5B, 5.4 h for LLaMA 8B, 10 h for Qwen 7B). The cost of constructing a PORTAL projector is negligible: the closed-form solution requires only forward passes on approximately 1,024 images and completes in under 30 seconds on a single RTX 3090.

Seeds. The cross-seed analysis in Section 3.2 compares two Stage-I projectors trained with seeds 42 and 120 on the same LCS-558k data. All other experiments use seed 42. Bootstrap confidence intervals are computed from B=1,000 resamples paired across methods within each (encoder, LLM) group.

Datasets and licenses. Table 5 lists the datasets used in our experiments.

C Evaluation Setup: Benchmarks and Protocol

All models are evaluated on 16 benchmarks grouped into four categories adapted from the Cambrian-1 [7] taxonomy (Knowledge, General, OCR, Vision), summarized in Table 6. Relative to the original Cambrian-1 suite, we replace OCRBench with OCRBench-v2 [56] and MMVP with BLINK [59]. All evaluations are performed using `lmms-eval` v0.4.1 [43].

Table 5: **Datasets used in our experiments.**

Dataset	Used for	License	Source
Training data			
LCS-558k	Stage-I pretrain	CC-BY-4.0	[2]
ShareGPT-1.2M	Stage-I pretrain	CC-BY-NC-4.0	[45]
LLaVA-665k	Stage-II finetune	CC-BY-4.0	[2]
Cambrian-737k	PORTAL image pool	Apache-2.0	[7]
COCO val2017	PORTAL image pool	CC-BY-4.0	[46]
Knowledge benchmarks			
AI2D	Evaluation	CC-BY-SA-4.0	[47]
ScienceQA	Evaluation	CC-BY-NC-SA-4.0	[48]
MathVista	Evaluation	CC-BY-SA-4.0	[49]
MMMU	Evaluation	Apache-2.0	[50]
General benchmarks			
GQA	Evaluation	CC-BY-4.0	[51]
MME	Evaluation	Research only	[52]
SeedBench	Evaluation	Apache-2.0	[53]
MMBench	Evaluation	CC-BY-4.0	[54]
OCR & Chart benchmarks			
TextVQA	Evaluation	CC-BY-4.0	[55]
OCRBench-v2	Evaluation	CC-BY-NC-4.0	[56]
DocVQA	Evaluation	Apache-2.0	[57]
ChartQA	Evaluation	GPL-3.0	[58]
Vision-Centric benchmarks			
CV-Bench	Evaluation	Apache-2.0	[7]
BLINK	Evaluation	Apache-2.0	[59]
RealWorldQA	Evaluation	CC-BY-ND-4.0	[60]

Table 6: **Evaluation benchmarks grouped by Cambrian-1 category.** ^{L} = Lite subset (500 samples).

Category	Benchmark	Samples	Metric
Knowledge	AI2D [47]	3,088	Exact match
	ScienceQA [48]	2,017	Exact match
	MathVista [49]	1,000	LLM-as-judge
	MMMU [50]	900	Accuracy
General	GQA [51]	12,578	Exact match
	MME [52]	2,374	Perception / 20
	SeedBench^{L} [53]	500	Accuracy (image)
	MMBench [54]	4,329	GPT-eval score
OCR	TextVQA [55]	5,000	Exact match
	OCRBench-v2 [56]	1,000	Accuracy
	DocVQA [57]	5,349	ANLS
	ChartQA [58]	2,500	Relaxed accuracy
Vision	CV-Bench-2D [7]	1,438	Accuracy
	CV-Bench-3D [7]	1,200	Accuracy
	BLINK [59]	1,437	Accuracy
	RealWorldQA [60]	765	Exact match

Scoring conventions. MME perception scores (out of 2,000) are divided by 20 to fit a 0–100 scale. MMBench reports GPT-eval scores (0–100) directly. MathVista uses GPT-4o as the judge model. All other metrics are reported as percentages (0–100).

Lite subsets. SeedBench uses the 500-sample Lite split from `lmms-eval`, denoted ^{L} in Table 1. Running the full SeedBench (17,990 samples) for every model configuration is computationally prohibitive.

Category averages. The Avg column in Table 1 is the arithmetic mean of all 16 benchmarks. Since each of the four categories contains four benchmarks, this is equivalent to the macro-average of the four category means.

Bootstrap confidence intervals. 95% confidence intervals for all category averages and overall scores are computed using B=1,000 bootstrap resamples [61]. Resampling is paired across methods within each (encoder, LLM) group: the same bootstrap indices are applied to all models that share a backbone, so per-sample correlations are preserved and the resulting intervals reflect method differences rather than sample noise. The %BL column reports the ratio of each model's overall score to its group baseline, bootstrapped under the same paired protocol.

C.1 Setup for the Data-Fraction and Interpolation Experiments

Throughout Section 3, we use the LLaVA 1.5 architecture. The vision encoder is CLIP ViT-L/14 (336px), the projector is a 2-layer MLP ($1024 \to 4096 \to 4096$), and the LLM is Vicuna 1.5 7B. The data-fraction experiment in Section 3.1 uses the smaller MobileLLaMA 1.4B backbone [62] with a $1024 \to 2048 \to 2048$ projector to keep the 11 data fractions on two corpora computationally feasible. Per-benchmark numbers for this experiment are reported in Appendix D.1. By default, we use LCS-558k (LAION/CC/SBU filtered pairs with BLIP captions) for Stage-I training and the standard LLaVA-665k instruction-tuning mix for Stage-II. Section 5 extends the picture to LLaMA 3.1 8B and Qwen 2.5 7B, and to SigLIP-SO400M as the vision encoder. For two corpora, LCS-558k and ShareGPT-1.2M, we train Stage-I projectors at evenly spaced data fractions (0 to 100%), apply identical LLaVA-665k instruction tuning on each, and evaluate on 16 benchmarks grouped into the four Cambrian-1 categories (Knowledge, General, OCR, Vision-Centric).

C.2 Cross-Seed vs. Cross-Data Interpolation

Section 3.2 reports that projectors trained on different datasets from the same initialization remain interpolable along the linear interpolation path. This appendix complements that result with a control that varies the random initialization instead of the training data, and contrasts the two cases.

Setup. We train two pairs of projectors using the LLaVA 1.5 setup described in Appendix C.1. *Cross-data* projectors share the same random initialization but are trained on different corpora (LCS-558k and ShareGPT-1.2M). *Cross-seed* projectors share the LCS-558k corpus but use two different random initializations. For each pair (W_A, W_B) we report the Frobenius distance $\|W_A - W_B\|_F$ and evaluate the linear interpolation $W_\alpha = (1-\alpha)\, W_A + \alpha\, W_B$ for $\alpha \in [0, 1]$. Stage-II tuning is identical for every W_α.

Results. The cross-seed pair is farther apart in weight space than the cross-data pair ($\|\Delta W\|_F = 92.4$ versus 63.7), but only the cross-data pair stays inside a low-loss region under interpolation. Cross-seed CIDEr [63] on the LLaVA captioning evaluation collapses from approximately 80 at the two endpoints to 0.7 at $\alpha = 0.5$. Cross-data projectors interpolate smoothly along the entire path and the aggregate downstream score peaks at $\alpha = 0.5$ (Figure 3).

Interpretation. Cross-seed projectors reach distinct loss basins separated by high-loss barriers, which is the standard picture for independently initialized networks [64]. Cross-data projectors trained from the same initialization remain inside one shared low-loss region. Initialization therefore selects the basin, and training data selects the particular solution reached within that basin. This is the property that PORTAL targets directly: the closed-form construction lands inside the same compatible region without using paired data.

D Full Downstream Benchmarks

D.1 LCS-558k Pretraining

Table 7: **Per-benchmark results for each LCS-558k data fraction.** Architecture: CLIP ViT-L/14 (336 px) and MobileLLaMA 1.4B [62], with a two-layer MLP projector ($1024 \to 2048 \to 2048$). Rows: fraction of the 558k-sample LCS corpus used for Stage-I pretraining (0% corresponds to no Stage-I, random projector init). Columns: 16 benchmarks grouped into four categories adapted from the Cambrian-1 taxonomy (Appendix C), followed by the overall average and its ratio to the 100% reference. All scores are normalized to 0–100 (MME is divided by 20, others are already percentages). ^{L}: 500-sample Lite subset from `lmms-eval` [43].

	Knowledge				General				OCR				Vision					
Pretrain	AI2D	SciQA	MathV	MMMU	GQA	MME	SEED^{L}	MMB	TxtVQA	OCR-Bv2	DocVQA	ChartQA	CV-2D	CV-3D	BLINK	RealWQ	**Avg**	**%BL**
0%	34.5	47.4	25.2	26.6	51.2	49.7	41.6	10.1	17.0	7.0	8.9	10.3	36.6	51.7	35.3	41.6	30.9	80.9
10%	37.5	53.0	24.3	28.0	56.5	56.4	58.6	39.3	32.1	7.8	13.1	10.6	43.7	53.2	35.7	44.4	37.1	97.1
20%	39.1	55.4	24.4	27.6	56.8	58.7	59.0	39.0	33.6	8.3	14.2	11.3	44.6	51.2	35.5	44.2	37.7	98.5
30%	39.1	53.7	25.8	29.3	56.6	57.6	58.6	37.7	34.8	7.6	14.9	10.6	44.6	51.8	35.7	44.3	37.7	98.5
40%	39.2	55.2	25.9	29.1	56.7	58.2	59.2	41.8	35.2	8.1	15.0	10.9	45.6	52.9	34.8	44.6	38.3	100.1
50%	39.7	54.9	25.8	29.2	57.1	58.2	59.6	42.2	35.5	8.2	15.4	10.9	46.0	53.2	35.6	46.3	**38.6**	**101.0**
60%	38.9	55.3	27.1	29.0	56.7	57.6	59.8	43.0	35.2	7.8	15.4	11.0	44.6	52.6	35.8	44.1	38.4	100.3
70%	39.4	55.2	27.1	29.0	56.9	58.7	59.6	41.2	35.6	7.6	15.2	11.5	45.5	52.4	34.6	45.2	38.4	100.5
80%	39.2	54.7	26.2	28.7	57.0	57.2	59.8	41.2	35.6	8.8	15.4	10.9	44.6	53.7	34.6	44.1	38.2	100.0
90%	39.4	55.2	26.4	28.8	57.4	57.0	59.4	41.4	35.7	7.8	15.5	11.1	46.2	54.0	34.7	44.2	38.4	100.4
100%	39.6	53.7	26.7	28.1	57.0	57.9	58.6	39.9	35.7	7.8	15.7	11.6	46.4	52.2	35.9	45.0	38.2	100.0

D.2 ShareGPT-1.2M Pretraining

Table 8: **Per-benchmark results for each ShareGPT-1.2M data fraction.** Layout matches Table 7 but with the ShareGPT-1.2M corpus used for Stage-I pretraining.

	Knowledge				General				OCR				Vision					
Pretrain	AI2D	SciQA	MathV	MMMU	GQA	MME	SEED^{L}	MMB	TxtVQA	OCR-Bv2	DocVQA	ChartQA	CV-2D	CV-3D	BLINK	RealWQ	**Avg**	**%BL**
0%	34.5	47.4	25.2	26.6	51.2	49.7	41.6	10.1	17.0	7.0	8.9	10.3	36.6	51.7	35.3	41.6	30.9	81.4
10%	38.4	53.7	24.3	28.1	55.1	54.5	53.5	50.0	29.6	6.9	12.5	11.2	40.8	52.4	35.2	43.1	36.8	97.0
20%	37.9	54.2	24.1	28.1	55.7	55.1	55.4	54.5	30.7	6.5	12.7	11.1	41.7	53.8	32.3	44.1	37.4	98.4
30%	38.7	53.9	24.4	28.3	56.4	55.7	56.4	51.5	31.1	6.8	13.1	11.2	40.5	52.9	35.6	45.2	37.6	99.0
40%	38.0	53.9	24.8	28.4	56.4	57.5	57.2	50.8	31.4	6.8	13.2	11.6	44.4	52.8	35.3	46.4	**38.1**	**100.2**
50%	38.3	54.4	26.0	27.7	56.7	57.0	54.9	52.3	31.9	7.3	14.0	11.6	41.5	52.8	34.6	48.2	38.1	100.2
60%	38.4	53.7	26.4	28.7	56.9	56.4	56.4	52.3	32.8	7.7	13.5	11.5	43.2	53.8	31.1	46.4	38.1	100.2
70%	38.3	54.4	26.8	28.4	57.1	56.3	55.6	50.8	33.9	7.8	14.1	11.4	45.0	52.2	30.9	45.4	38.0	100.1
80%	38.7	53.7	26.0	28.6	57.1	57.3	55.4	49.2	34.1	7.9	14.1	11.5	41.9	52.2	30.9	46.8	37.8	99.6
90%	38.2	54.9	26.2	29.2	57.1	57.1	57.2	50.0	34.1	8.2	14.4	11.1	43.0	51.9	24.2	45.1	37.6	99.1
100%	38.6	54.4	25.5	28.2	57.2	55.8	57.0	53.0	34.4	7.9	14.5	11.9	42.8	53.0	26.7	46.7	38.0	100.0

E Additional Experiments

In this section, we provide additional experiments that further evaluate the scope and robustness of PORTAL. We study whether the coarse-alignment phenomenon extends to a substantially higher-capacity connector, whether PORTAL remains effective under larger-scale multimodal instruction tuning, how its performance varies across task categories, how it compares with alternative unpaired alignment methods, and whether training-free alignment affects object hallucination.

E.1 Comparison with Alternative Unpaired Alignment Methods

Random initialization in our main experiments serves as a control for completely skipping Stage-I alignment rather than as the only competing alignment method. We additionally compare PORTAL against CORAL [26], which aligns second-order feature statistics in closed form, and Gromov-Wasserstein (GW) alignment [25], an iterative unpaired alignment method.

Table 9: Comparison with alternative alignment methods. All methods except the conventional Stage-I baseline require no paired image-caption samples.

Initialization	Avg16	% of Baseline
Stage-I trained baseline	48.03	100.0
Random	44.90	93.5
CORAL	46.36	96.5
Gromov-Wasserstein	47.06	98.0
PORTAL	**47.43**	**98.8**

Table 10: Characteristics of the unpaired initialization methods considered in Table 9.

Method	Paired Samples	Projector Training	Construction
Random	No	No	Random initialization
CORAL	No	No	Closed form
GW	No	No	Iterative optimization
PORTAL	No	No	Closed form

Both CORAL and GW substantially improve over random initialization, recovering 96.5% and 98.0% of the paired Stage-I baseline, respectively (Table 9). This confirms more broadly that useful projector initialization can be obtained from unpaired feature distributions. PORTAL performs best among the unpaired methods considered here, recovering 98.8% of the trained baseline while retaining a closed-form construction and requiring neither paired samples nor projector optimization.

E.2 Performance Retention Across Task Categories

We next examine whether the aggregate performance of PORTAL hides systematic differences across task categories. Table 11 reports performance relative to the corresponding conventionally Stage-I-trained baseline for Knowledge, General, OCR, and Vision-Centric benchmarks.

Table 11: PORTAL performance as a percentage of the corresponding Stage-I-trained baseline, broken down by task category. The last row averages the retention rate across the seven (vision encoder, LLM) configurations.

LLM	Vision Encoder	Knowledge	General	OCR	Vision	Overall
Vicuna 1.5 7B	CLIP	100.3	97.3	96.6	100.5	98.8
Vicuna 1.5 7B	SigLIP	100.6	98.1	97.6	101.0	99.3
LLaMA 3.1 8B	CLIP	103.6	100.7	103.9	103.8	102.7
LLaMA 3.2 1B	CLIP	102.5	109.8	112.1	103.1	106.2
Qwen 2.5 7B	CLIP	98.5	99.4	92.6	99.3	98.3
Qwen 2.5 0.5B	CLIP	100.9	113.2	121.1	97.9	106.1
OLMo 2 1B	CLIP	98.2	95.4	92.9	97.4	96.3
Mean across configurations	—	**100.7**	**102.0**	**102.4**	**100.4**	**101.1**

On average across configurations, PORTAL retains 100.7%, 102.0%, 102.4%, and 100.4% of the trained-baseline performance on Knowledge, General, OCR, and Vision-Centric tasks, respectively. Thus, the aggregate performance of PORTAL is not driven by a single task category. At the same time, paired Stage-I training remains beneficial for particular encoder-LLM combinations and individual benchmarks. For example, the largest category-level deficits occur for OCR under Qwen-2.5-7B and OLMo-2-1B. The results therefore suggest a more nuanced picture than fine-grained tasks universally requiring paired Stage-I supervision: the value of paired supervision depends on the specific backbone, encoder, and benchmark.

E.3 Large-Scale Multimodal Instruction Tuning

Our main experiments use LLaVA-665k for Stage-II instruction tuning. We further test whether PORTAL remains effective under a substantially larger multimodal fine-tuning regime using 3.2M samples subsampled from Cambrian-7M. This setting provides a stronger test of whether PORTAL initialization remains stable when followed by heavier multimodal training.

Table 12: Performance under large-scale Stage-II multimodal instruction tuning with 3.2M samples. The confidence interval reports the 95% CI for the difference between PORTAL and the Stage-I-trained baseline.

Category	Baseline	PORTAL	Δ	95% CI
Knowledge	45.95	46.68	+0.72	$[-0.40, +1.88]$
General	59.60	59.88	+0.27	$[-0.71, +1.27]$
OCR	30.28	31.45	+1.18	$[+0.67, +1.68]$
Vision	50.12	51.28	+1.16	$[-0.13, +2.46]$
Overall	**46.49**	**47.32**	**+0.83**	$[\mathbf{+0.36}, \mathbf{+1.35}]$

PORTAL remains effective under this substantially heavier Stage-II training regime (Table 12). Across the 16 benchmarks, the overall mean increases from 46.49 with conventional Stage-I initialization to 47.32 with PORTAL, corresponding to a statistically significant improvement of $+0.83$ points with a 95% confidence interval of $[+0.36, +1.35]$. Importantly, PORTAL does not lead to performance collapse or degradation under large-scale multimodal fine-tuning. The result suggests that the benefit of the closed-form initialization is not restricted to the standard LLaVA-665k Stage-II regime.

E.4 Object Hallucination

Finally, we evaluate whether replacing paired Stage-I alignment with coarse statistical alignment increases object hallucination. We evaluate all seven (vision encoder, LLM) configurations on POPE [44], a standard benchmark for object hallucination (Table 13). PORTAL matches or outperforms the conventionally Stage-I-trained baseline across all seven configurations, with improvements ranging from $+0.2$ to $+2.2$ F1 points. Thus, replacing paired Stage-I projector training with PORTAL does not increase object hallucination in these experiments and can slightly improve hallucination robustness.

Table 13: POPE F1 scores across all seven (vision encoder, LLM) configurations. Numbers in parentheses indicate the improvement of PORTAL over the conventionally Stage-I-trained baseline.

LLM	Vision Encoder	Baseline	Random	PORTAL
Vicuna 1.5 7B	CLIP	85.5	85.3	**85.7** $(+0.2)$
Qwen 2.5 7B	CLIP	85.7	85.5	**86.4** $(+0.7)$
LLaMA 3.1 8B	CLIP	84.0	84.2	**84.9** $(+0.9)$
Vicuna 1.5 7B	SigLIP	84.6	79.1	**85.1** $(+0.5)$
OLMo 2 1B	CLIP	85.3	84.8	**85.5** $(+0.2)$
LLaMA 3.2 1B	CLIP	84.1	84.0	**86.2** $(+2.1)$
Qwen 2.5 0.5B	CLIP	82.9	82.7	**85.1** $(+2.2)$

E.5 Higher-Capacity Connector

Our main experiments use either a two-layer MLP or a linear projector. A possible alternative explanation for the observed coarse-alignment behavior is therefore that it arises from limited projector capacity. To test this, we additionally evaluate PORTAL with the C-Abstractor connector from Honeybee [12], a substantially higher-capacity architecture that performs both token mixing and sequence compression.

In the LLaMA-3.2-1B + CLIP setting, C-Abstractor contains 120.2M parameters, approximately $19\times$ more than the 6.3M-parameter two-layer MLP, and compresses the visual sequence from 576 to 144 tokens. We initialize the residual branches of C-Abstractor to zero such that the connector initially reduces to applying PORTAL to the pooled visual features. All remaining C-Abstractor parameters are then optimized during the same Stage-II training used for the other initialization schemes.

As shown in Table 14, PORTAL reaches an average score of 42.93, compared with 42.98 for conventional Stage-I training, while direct Stage-II training from random initialization reaches only 41.46. PORTAL also outperforms the Stage-I-trained baseline on 7 of the 16 individual benchmarks. Thus, increasing connector capacity by approximately $19\times$, while additionally enabling token mixing and sequence compression, does not yield an aggregate advantage for paired Stage-I training over PORTAL. Together with our linear and two-layer MLP experiments, this result provides evidence that the observed coarse-alignment behavior cannot be explained simply by insufficient projector capacity. We do not, however, claim that every possible connector architecture must exhibit identical behavior.

Table 14: Performance with the higher-capacity C-Abstractor connector. Despite substantially increasing connector capacity and introducing explicit token mixing and sequence compression, PORTAL remains essentially indistinguishable from conventional Stage-I training.

Task	**Baseline**	**Random**	**PORTAL**
AI2D	49.3	46.7	48.0
ScienceQA	68.2	65.2	66.2
MathVista	26.9	25.9	26.2
MMMU	32.3	34.6	33.9
GQA	60.6	59.7	60.7
MME-P	60.8	59.6	65.4
SEED-L	58.4	54.7	56.2
MMBench	58.4	52.4	54.9
TextVQA-OCR	46.7	43.6	45.4
OCRBench-v2	10.7	11.2	11.5
DocVQA	13.2	11.8	12.7
ChartQA	11.1	10.4	10.8
CV-Bench 2D	50.1	47.8	50.8
CV-Bench 3D	54.2	53.9	55.2
BLINK	38.3	37.0	40.9
RealWorldQA	48.6	48.9	48.1
Average	**42.98**	**41.46**	**42.93**

F PORTAL Construction: Pseudocode

Listing 1 states the PORTAL construction as a self-contained PyTorch routine. The function consumes a sample of source features (*e.g.*, CLIP patch tokens), a sample of target features (*e.g.*, the LLM's input embedding table or the hidden states after block ℓ), and a rank k. It returns the parameters $(\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2)$ of the standard two-layer LLaVA projector $f(\mathbf{x}) = \mathbf{W}_2\,\phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$ from Section 4, with ϕ a GELU. The first layer is the closed-form Gaussian optimal-transport map of Eq. (2), and the second layer is fixed to the identity. No paired captions and no Stage-I optimization are required, only one PCA on each side.

```python
import torch

def portal(X, Y, k=512):
    """
    X: source features, shape (N_s, d_s) -- e.g. CLIP patch tokens.
    Y: target features, shape (N_t, d_t) -- LLM input embeddings,
                                        or post-block-l hidden
                                        states for PORTAL-ctx-l.
    k: PCA rank for the shared subspace.

    Returns (W1, b1, W2, b2) for the LLaVA projector
        f(x) = W2 @ gelu(W1 @ x + b1) + b2.
    """
    # 1. Source statistics: top-k PCA on centered source features.
    mu_s = X.mean(dim=0) # (d_s,)
    _, S_s, Vt_s = torch.linalg.svd(X - mu_s, full_matrices=False)
    V_s = Vt_s[:k].T # (d_s, k)
    sigma_s = S_s[:k] / (X.shape[0] - 1) ** 0.5 # (k,)

    # 2. Target statistics: top-k PCA on centered target features.
    mu_t = Y.mean(dim=0) # (d_t,)
    _, S_t, Vt_t = torch.linalg.svd(Y - mu_t, full_matrices=False)
    V_t = Vt_t[:k].T # (d_t, k)
    sigma_t = S_t[:k] / (Y.shape[0] - 1) ** 0.5 # (k,)

    # 3. Closed-form projector (Eq. 2): W2 fixed to identity.
    W1 = V_t @ torch.diag(sigma_t / sigma_s) @ V_s.T # (d_t, d_s)
    b1 = mu_t - W1 @ mu_s # (d_t,)
    W2 = torch.eye(Y.shape[1])
    b2 = torch.zeros(Y.shape[1])
    return W1, b1, W2, b2
```

Listing 1: PORTAL: training-free projector via Gaussian optimal transport. The same routine yields PORTAL-ctxℓ when Y is replaced by post-block-ℓ hidden states (Appendix G).

Inputs. The source matrix `X` contains all per-patch tokens from a small unlabeled image pool: with P patches per image and N_{img} images, $N_s = N_{\text{img}} \cdot P$. Throughout the paper we use a default of $N_{\text{img}} = 1024$ images from COCO `val2017` (Appendix I). For default PORTAL, the target matrix `Y` is the LLM's input embedding table, with one row per vocabulary token. For PORTAL-ctxℓ, `Y` instead holds the hidden states after block ℓ, which we collect by passing a list of English words and short phrases (the `concepts.txt` file of LatentLens) through the LLM (Appendix G).

Interpretation. Read right to left, `V_s.T` projects each centered source token onto the source's top-k principal axes, the diagonal rescaling `diag(sigma_t / sigma_s)` rescales each axis from source to target spread, and `V_t` rotates the result into the target's top-k principal frame. Adding `b1 = mu_t - W1 @ mu_s` then recenters to the target mean. The composition matches first and second moments of source and target on the k-dimensional shared subspace. We use k=512 as the default (Appendix K). The construction is well-defined for any $k \le \min(d_s, d_t, N_s, N_t)$.

Cost. The procedure runs two thin SVDs and two matrix multiplies. For the default setting ($N_s \approx 1024 \cdot 576$, $d_s = 1024$, $d_t = 4096$, $k = 512$) the construction completes in under 30 seconds on a single RTX 3090 (Appendix B) and uses no captions, no paired data, and no gradient steps. This stands in contrast to standard Stage-I projector training, which consumes a paired image-caption corpus (typically LCS-558k) and about 2,200 optimizer steps (558k pairs at batch size 256).

G Contextualized Targets: Methodology Details

This appendix provides the implementation details behind PORTAL-ctxℓ (Section 4).

Probe corpus. We use the `concepts.txt` corpus released with LatentLens [21], a list of English concepts originally assembled for probing latent directions in LLMs. The corpus offers broad semantic coverage and is not tied to any downstream task, which makes it a representative sample of the input distribution of the LLM at depth ℓ.

Cached hidden states. For LLMs covered by LatentLens (LLaMA 3.1 8B, Qwen 2.5 7B, Gemma 2 9B) we use the pre-computed hidden states distributed on Hugging Face. For each LLM and each layer ℓ, the cache stores the post-block-ℓ hidden state of every token in every `concepts.txt` entry, collected in a single tensor $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N_t \times d_t}$. The available layers are $\ell \in \{1, 4, 8, 12, 16, 20\}$. For Vicuna v1.5, which is not part of the LatentLens release, we run `concepts.txt` through the model and store the corresponding hidden states in the same format. The tensor $\mathbf{H}^{(\ell)}$ replaces the LLM's input embedding table as the target distribution in Eq. (2). The remainder of the PORTAL construction is unchanged.

Layer indexing. ℓ denotes the residual stream *after* the ℓ-th transformer block (1-indexed), following the LatentLens convention. The setting ℓ=0 recovers standard PORTAL targeting the LLM's input embedding table.

H Cosine Dispersion: Theory and Empirical Checks

This section consolidates the analysis underlying Section 4.2. It quantifies how non-Gaussian the relevant source and target distributions are (Appendix H.1), proves the spectrum-to-ω bound (Appendix H.2), reports the empirical check of that bound on PORTAL projector outputs (Appendix H.3), and extends the empirical check to the full MLP-GELU projector (Appendix H.4).

H.1 Distributional Audit of Source and Target Modalities

PORTAL treats both modalities as Gaussian in Eq. (2) only to obtain a closed-form transport map. The empirical source and target distributions are not exactly Gaussian. This subsection quantifies the deviation so that the reader can assess how reliable the second-order summary used by PORTAL is.

Procedure. For each distribution we draw n=5,000 tokens (CLIP and SigLIP patches from COCO `val2017` images, rows of the LLM's input embedding table), project the centered features onto their top k=128 principal components and whiten the resulting coordinates to unit per-axis variance. On these coordinates we compute four statistics. Mardia's multivariate skewness b_1 [65], reported as b_1/k^3, which is close to zero under Gaussianity. Mardia's kurtosis b_2 [65], reported as the excess $b_2/k^2 - (1 + 2/k)$ over its Gaussian value, which quantifies tail weight relative to a Gaussian. The fraction of the k whitened marginals for which the Anderson-Darling test [66] rejects $\mathcal{N}(0, 1)$ at the 5% level. The effective dimension $d_{\text{eff}} = (\operatorname{tr} \mathbf{\Sigma})^2 / \operatorname{tr}(\mathbf{\Sigma}^2)$ [67], which summarizes how variance is distributed across the leading directions. Results are reported in Table 15.

Table 15: **Distributional audit on top k=128 principal components.** The columns most relevant to the argument in Section 4 are d_{eff} and Mardia b_1. The Anderson-Darling rejection rate (AD rej.) and the excess kurtosis identify where the deviation from Gaussianity is concentrated.

Type	Distribution	n	d	d_{eff}	Mardia b_1	Exc. kurt.	AD rej.
Source	CLIP ViT-L/14 patches	5,000	1,024	4.5	0.0022	0.487	100%
Source	SigLIP SO400M patches	5,000	1,152	50.9	0.0026	0.278	100%
Target	Vicuna 7B embedding table	5,000	4,096	1,429.6	0.0005	0.071	84%
Target	LLaMA 3.1 8B embedding table	5,000	4,096	1,713.2	0.0005	0.131	98%
Target	Qwen 2.5 7B embedding table	5,000	3,584	830.1	0.0004	0.114	100%

Results. No measured distribution is exactly Gaussian, and the Anderson-Darling test rejects Gaussianity on almost all top marginals. This is expected for high-dimensional learned features and does not, on its own, imply that PORTAL should fail.

The question is where the deviation lives. Skewness is small for every distribution ($b_1/k^3 \leq 0.003$), so the residual deviation from Gaussianity lies in tail weight. The tails are heaviest for the vision features (excess kurtosis 0.28 to 0.49) and light for the LLM input embedding tables (0.07 to 0.13).

A second question is whether the distributions that PORTAL targets spread their variance over many directions. The input embedding tables do, with d_{eff} between 830 and 1,713, while the CLIP patches concentrate their variance in a few directions ($d_{\text{eff}} = 4.5$).

H.2 Proof of Proposition 1

Proposition 1 states that the variance of the cosine between two random tokens is close to $1/d_{\text{eff}}(\mathbf{\Sigma})$ when token lengths vary little. The proof first computes the second moment of the plain dot product $\mathbf{z}^\top \mathbf{z}'$, which depends only on $\mathbf{\Sigma}$, and then bounds how much dividing by the two token lengths can change it.

Notation. The trace $\operatorname{tr} \mathbf{\Sigma}$ equals the mean squared length $\mathbb{E}[\|\mathbf{z}\|^2]$, because $\mathbf{z}$ has mean zero. We write

- $\mathbf{u} = \mathbf{z}/\|\mathbf{z}\|$ and $\mathbf{u}' = \mathbf{z}'/\|\mathbf{z}'\|$ for the unit vectors along $\mathbf{z}$ and $\mathbf{z}'$,
- $c = \mathbf{u}^\top \mathbf{u}' = \cos \angle(\mathbf{z}, \mathbf{z}')$ for the cosine,
- $a = \|\mathbf{z}\|^2/\operatorname{tr} \mathbf{\Sigma}$ and $a' = \|\mathbf{z}'\|^2/\operatorname{tr} \mathbf{\Sigma}$ for the squared lengths relative to their mean, so that $\mathbb{E}[a] = 1$ and $\operatorname{std}(a) = \kappa$,
- $m = \mathbb{E}|1-a|$ for the mean deviation of a from 1. By the Cauchy-Schwarz inequality, $m \le \sqrt{\mathbb{E}[(1-a)^2]} = \kappa$.

Proof. Step 1: the dot product. Averaging over $\mathbf{z}$ with $\mathbf{z}'$ held fixed, and using $\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{\Sigma}$,

$$\mathbb{E}\big[(\mathbf{z}^\top \mathbf{z}')^2\big] = \mathbb{E}\big[\mathbf{z}'^\top \mathbf{z}\, \mathbf{z}^\top \mathbf{z}'\big] = \mathbb{E}\big[\mathbf{z}'^\top \mathbf{\Sigma}\, \mathbf{z}'\big] = \operatorname{tr}\big(\mathbf{\Sigma}\, \mathbb{E}[\mathbf{z}'\mathbf{z}'^\top]\big) = \operatorname{tr}(\mathbf{\Sigma}^2).$$

For the normalized squared dot product $q = (\mathbf{z}^\top \mathbf{z}')^2/(\operatorname{tr} \mathbf{\Sigma})^2$ this gives

$$\mathbb{E}[q] = \frac{\operatorname{tr}(\mathbf{\Sigma}^2)}{(\operatorname{tr} \mathbf{\Sigma})^2} = \frac{1}{d_{\text{eff}}(\mathbf{\Sigma})}.$$

This step is exact and holds for any distribution.

Step 2: from the dot product to the cosine. The cosine is the dot product divided by both lengths, so $c^2 = q/(aa')$, or $q = c^2 aa'$. Hence

$$\mathbb{E}[c^2] - \frac{1}{d_{\text{eff}}(\mathbf{\Sigma})} = \mathbb{E}[c^2 - q] = \mathbb{E}\big[c^2(1 - aa')\big].$$

Since $c^2 \le 1$, the absolute value of the right side is at most $\mathbb{E}|1 - aa'|$. Writing $1 - aa' = (1-a) + (1-a') - (1-a)(1-a')$, the triangle inequality and the independence of a and a' give

$$\mathbb{E}|1 - aa'| \le \mathbb{E}|1-a| + \mathbb{E}|1-a'| + \mathbb{E}|1-a|\,\mathbb{E}|1-a'| = 2m + m^2.$$

Step 3: the mean cosine. The variance subtracts the squared mean, $\operatorname{Var}(c) = \mathbb{E}[c^2] - \mathbb{E}[c]^2$, so it remains to bound $\mathbb{E}[c]$, the average cosine between two independent tokens. By independence, $\mathbb{E}[c] = \mathbb{E}[\mathbf{u}]^\top \mathbb{E}[\mathbf{u}'] = \|\mathbb{E}[\mathbf{u}]\|^2$. Because $\mathbf{z} = \sqrt{\operatorname{tr} \mathbf{\Sigma}}\, \sqrt{a}\, \mathbf{u}$ has mean zero, $\mathbb{E}[\sqrt{a}\, \mathbf{u}] = \mathbf{0}$, and therefore

$$\mathbb{E}[\mathbf{u}] = \mathbb{E}\big[(1 - \sqrt{a})\, \mathbf{u}\big], \qquad \|\mathbb{E}[\mathbf{u}]\| \le \mathbb{E}|1 - \sqrt{a}| \le \mathbb{E}|1-a| = m,$$

where we used $\|\mathbf{u}\| = 1$ and $|1 - \sqrt{a}| = |1-a|/(1+\sqrt{a}) \le |1-a|$. Hence $\mathbb{E}[c]^2 = \|\mathbb{E}[\mathbf{u}]\|^4 \le m^4$.

Combining the steps. By Steps 2 and 3 and $m \le \kappa$,

$$\left|\operatorname{Var}(c) - \frac{1}{d_{\text{eff}}(\mathbf{\Sigma})}\right| \le \left|\mathbb{E}[c^2] - \frac{1}{d_{\text{eff}}(\mathbf{\Sigma})}\right| + \mathbb{E}[c]^2 \le 2m + m^2 + m^4 \le 2\kappa + \kappa^2 + \kappa^4. \qquad \square$$

The quantity ω^2 in Eq. (3) is the empirical variance of the cosines over all pairs of n sampled tokens. Both of its pairwise averages are averages of a bounded function over all pairs (U-statistics), so ω^2 converges to $\operatorname{Var}(c)$ as n grows.

H.3 Cosine Dispersion of PORTAL Projector Outputs

This appendix measures ω on the PORTAL outputs for the three CLIP pairs with a 7B or 8B LLM in Table 1. We construct PORTAL as in Appendix F from the first 1024 COCO `val2017` images, apply it to the next 100 images (576 CLIP patches per image), and report two quantities. The intra-image ω is averaged across images, with each image contributing the dispersion of its own 576 output tokens. The population ω is computed over 576 output tokens sampled at random across all images. The intra-image quantity reflects the dispersion seen by attention on a single forward pass. The population quantity is the distributional property that Proposition 1 describes.

Table 16: **Cosine dispersion ω of projector outputs and downstream average.** For each (encoder, LLM) pair, *Source* is ω of the raw source patches, *Random* and *PORTAL* are ω after applying the corresponding projector, and *Target* is ω of the LLM's input embedding table restricted to the top-512 principal subspace that PORTAL matches. Random and PORTAL denote the first layer $\mathbf{W}_1\mathbf{x} + \mathbf{b}_1$ of each projector. *PORTAL* d_{eff} reports the effective dimension of the PORTAL output cloud. Downstream Avg columns from Table 1.

Encoder	LLM	Scale	Source	Random	PORTAL	Target	PORTAL d_{eff}	Random Avg	PORTAL Avg	Trained Avg
CLIP	Vicuna 7B	pop	0.234	0.234	**0.100**	0.056	266.7	44.9	**47.4**	48.0
		intra	0.252	0.251	0.235	0.056	266.7	44.9	**47.4**	48.0
CLIP	LLaMA 3.1 8B	pop	0.234	0.234	**0.101**	0.057	257.4	50.4	**52.2**	50.8
		intra	0.252	0.251	0.234	0.057	257.4	50.4	**52.2**	50.8
CLIP	Qwen 2.5 7B	pop	0.234	0.234	**0.094**	0.054	289.3	48.6	**51.7**	52.6
		intra	0.252	0.251	0.245	0.054	289.3	48.6	**51.7**	52.6

PORTAL reduces ω from the source value of 0.23 to between 0.09 and 0.10, towards the target value of 0.05 to 0.06. Over the full embedding table, the target's ω is 0.02 to 0.03. A random projector roughly preserves angles and therefore leaves ω at the source value. The intra-image ω does not separate the three projectors, because patches from a single image share a common offset that no global linear map can remove.

The measurement is correlative. PORTAL moves ω most of the way from the source towards the target, and we do not claim that ω is the sole geometric property responsible for the downstream performance of PORTAL.

H.4 Post-Projection Statistics: Linear vs. MLP-GELU Output

The closed-form construction in Eq. (2) gives an affine map $\mathbf{W}_1\mathbf{x} + \mathbf{b}_1$ that matches the target's first and second moments under a Gaussian approximation. The implemented projector then passes that affine output through GELU. This subsection reports distributional statistics of the actual projector output for each (encoder, LLM) pair across four constructions. *Random+GELU* is a Random Kaiming initialization followed by GELU. *PORTAL (preact)* is the PORTAL preactivation alone, which is the linear object the theory in Section 4.2 applies to. *PORTAL+GELU* is the PORTAL preactivation followed by GELU, which is the implemented PORTAL projector. *Trained Stage-I* is a trained Stage-I projector loaded from checkpoint. The target row reports the same statistics on the LLM's input embedding table, restricted to the top-512 principal subspace that PORTAL matches. *Trained Stage-I* is the public LLaVA-1.5 projector for Vicuna and our trained projectors for LLaMA and Qwen. For the latter two we report only the statistics that do not depend on the target.

For each output cloud we report five statistics. The mean offset $\|\Delta\boldsymbol{\mu}\|$ is the L^2 distance between the output mean and the target mean, with a relative version normalized by $\|\boldsymbol{\mu}_t\|$. The spectrum error is the relative L^2 distance between the top-128 output and target singular values. The effective dimension is $d_{\text{eff}} = (\operatorname{tr}\boldsymbol{\Sigma})^2/\operatorname{tr}(\boldsymbol{\Sigma}^2)$. The norm coefficient of variation $\text{cv} = \text{std}(\|\mathbf{z}\|)/\mathbb{E}[\|\mathbf{z}\|]$ serves as a concentration check. The population pairwise-cosine standard deviation is ω as defined in Eq. (3). Statistics are computed over 100 COCO `val2017` images of 576 CLIP patches each, with PORTAL constructed from a disjoint set of 1024 images.

The table supports four observations. First, Random+GELU is far from the target on every column, consistent with the SigLIP collapse reported in Table 1. Second, the PORTAL preactivation matches the target mean and top-128 spectrum, comes close to its d_{eff}, and moves ω from 0.23 at the source to about 0.10, against 0.05 to 0.06 for the target. Third, applying GELU to the PORTAL preactivation roughly preserves d_{eff} and ω, even though the spectrum error grows by an order of magnitude. This empirical observation closes the gap between the theory of the linear object and the implemented MLP. Fourth, a trained Stage-I projector deviates substantially from the target distribution on every column we report. Despite this, trained Stage-I and PORTAL reach comparable downstream Avg in Table 1. The two solutions are distributionally distinct, which is consistent with the broad-solution-space rather than with the view that Stage-I learns a unique target distribution.

I Image Pool for PORTAL Construction

PORTAL uses an unlabeled image pool only to estimate the CLIP patch-feature statistics required by Eq. (2), and requires neither captions nor paired text. We assess the sensitivity of the constructed projector to the identity and size of this pool. Table 18 reports an ablation on LLaMA 3.2 1B/CLIP along two axes.

- **Source diversity.** COCO (`val2017`, single domain) vs. a reservoir-sampled subset of Cambrian-737k [7] spanning nine sources (`coco`, `vg`, `ocr_vqa`, `gqa`, `chartqa`, `textvqa`, `ai2d`, `docvqa`, `dvqa`). For reference, the trained baseline uses the LCS-558k [2] caption pool.
- **Pool size.** 1024 images (standard) vs. 4096 images.

Table 17: **Post-projection statistics on CLIP patch tokens.** Distributional fit to the target distribution for four projector constructions, plus a target row for reference. Smaller $\|\Delta\boldsymbol{\mu}\|$ and spectrum error indicate a closer fit to the target. The d_{eff} and ω values should be read against the target row of the same LLM.

LLM	Method	$\|\Delta\boldsymbol{\mu}\|$	Rel.	Spec. err.	d_{eff}	cv	ω (pop)
Vicuna 7B	*target*	0.000	0.000	0.000	319.7	0.123	0.056
	Random+GELU	7.144	47.33	24.25	3.6	0.682	0.212
	PORTAL (preact)	0.021	0.142	0.066	266.7	0.281	0.100
	PORTAL+GELU	0.077	0.508	0.470	266.9	0.281	0.100
	Trained Stage-I	33.18	219.8	142.4	2.4	1.047	0.232
LLaMA 3.1 8B	*target*	0.000	0.000	0.000	308.2	0.135	0.057
	Random+GELU	7.145	83.28	47.07	3.6	0.682	0.212
	PORTAL (preact)	0.011	0.133	0.088	257.4	0.279	0.101
	PORTAL+GELU	0.043	0.505	0.460	257.3	0.279	0.101
	Trained Stage-I	–	–	–	2.4	1.019	0.239
Qwen 2.5 7B	*target*	0.000	0.000	0.000	352.5	0.216	0.054
	Random+GELU	6.791	81.61	25.55	3.6	0.686	0.212
	PORTAL (preact)	0.020	0.242	0.097	289.3	0.294	0.094
	PORTAL+GELU	0.043	0.511	0.456	289.1	0.294	0.094
	Trained Stage-I	–	–	–	2.0	1.147	0.221

The smaller single-domain COCO pool (1024 images) matches or exceeds the larger Cambrian pool (4096 images) on 12 of 15 benchmarks, with 2 exact ties and 3 marginal Cambrian wins (AI2D, MMMU, DocVQA). The COCO pool yields an aggregate gain of $+0.6$ Avg (44.0 vs. 43.4) and $+1.3$ %BL (106.1 vs. 104.8).

These numbers indicate that the second-order statistics of a generic natural-image distribution are sufficient to specify the whitening map. Increasing source diversity provides no additional benefit once the statistics have converged, in agreement with the rapid decay of the leading singular values of a CLIP feature matrix. On this configuration, PORTAL is insensitive to the choice between the two pools.

Table 18: **Image-pool ablation for the PORTAL construction, per-benchmark breakdown.** Setup: LLaMA 3.2 1B/CLIP, Stage-II on LLaVA-665k. Both PORTAL variants use rank k=512 and require no paired captions. **Bold**: best between the two PORTAL rows per benchmark. Avg and %BL are computed over the 15 benchmarks shown.

			Knowledge				General				OCR			Vision					
Pool	Src	#Img	AI2D	SciQA	MathV	MMMU	GQA	MME	SEED^I	MMB	OCR-Bv2	DocVQA	ChartQA	CV-2D	CV-3D	BLINK	RealWQ	Avg	%BL
Baseline†	LCS	558k	46.5	66.2	26.9	33.6	59.7	58.4	56.2	46.1	10.1	12.8	10.8	53.5	56.4	37.2	47.3	41.4	100.0
Random	–	0	46.6	66.9	26.3	34.2	59.8	58.1	56.0	45.2	10.4	12.8	11.0	54.2	54.9	37.4	46.9	41.4	99.8
PORTAL	Cambrian	4096	**48.2**	66.7	26.9	**34.3**	62.0	64.3	**59.6**	53.6	10.8	**16.8**	**12.0**	53.5	55.8	37.7	49.4	43.4	104.8
PORTAL	COCO	1024	47.6	**67.1**	**29.2**	33.7	**62.1**	**66.0**	**59.6**	**54.2**	**11.4**	16.3	**12.0**	**54.6**	**57.3**	**38.9**	**49.7**	**44.0**	**106.1**

J Projector Architecture: Linear vs. MLP

The standard LLaVA projector is a two-layer MLP with a GELU activation. We test whether the benefit of PORTAL also transfers to a simpler architecture. We replace the MLP with a single linear layer (no nonlinearity, no hidden expansion, input dimension 1024, output dimension d_{LLM}) and keep all other components fixed: LLaMA 3.2 1B/CLIP, Stage-II tuning on LLaVA-665k, identical hyperparameters, and the same PORTAL construction with rank k=512 on the COCO-1024 image pool from Appendix I. For each projector we evaluate three initialization schemes: a trained baseline (Stage-I on LCS-558k), Random init, and PORTAL. The MLP rows are reproduced from Table 1.

Table 19 reports the per-benchmark breakdown. Under the MLP, Random init matches the trained baseline (41.5 vs. 41.6 Avg, 99.9%) and PORTAL reaches 44.2 Avg (106.2%). Under the linear projector, Random init falls to 37.6 Avg (94.2% of the linear baseline) and PORTAL reaches 44.7 Avg (111.8%), the highest macro average among the six configurations. The Random-to-PORTAL gap is 6.3 percentage points under the MLP and 17.6

under the linear projector. PORTAL is the stronger zero-shot init on 15 of 16 benchmarks under the MLP and on 16 of 16 benchmarks under the linear projector. This supports the interpretation that the gain from PORTAL is a property of the projector initialization and transfers across projector architectures.

Table 19: **Linear versus MLP-2x-GELU projector, per-benchmark breakdown.** Setup: LLaMA 3.2 1B/CLIP, Stage-II on LLaVA-665k. Each architecture is anchored to its own trained baseline for %BL. **Bold**: best between Random and PORTAL within an architecture. Underline: second best.

		Knowledge				General				OCR				Vision					
Projector	**Init**	AI2D	SciQA	MathV	MMMU	GQA	MME	SEED^I	MMB	TxtVQA	OCR-Bv2	DocVQA	ChartQA	CV-2D	CV-3D	BLINK	RealWQ	**Avg**	**%BL**
MLP-2x-GELU	Baseline†	46.5	66.2	26.9	33.6	59.7	58.4	56.2	46.1	43.7	10.1	12.8	10.8	53.5	56.4	37.2	47.3	41.6	100.0
	Random	46.6	66.9	26.3	**34.2**	59.8	58.1	56.0	45.2	43.8	10.4	12.8	11.0	54.2	54.9	37.4	46.9	41.5	99.9
	PORTAL	**47.6**	**67.1**	**29.2**	33.7	**62.1**	**66.0**	**59.6**	**54.2**	**47.1**	**11.4**	**16.3**	**12.0**	**54.6**	**57.3**	**38.9**	**49.7**	**44.2**	**106.2**
Linear	Baseline†	46.5	65.5	27.1	33.3	54.7	53.1	47.3	49.8	42.6	10.1	11.0	10.0	50.1	54.4	36.8	46.7	39.9	100.0
	Random	46.0	63.2	24.6	30.9	53.0	52.8	46.1	43.8	36.8	7.9	7.3	8.8	45.6	52.8	37.7	44.4	37.6	94.2
	PORTAL	**48.2**	**68.5**	**28.2**	**34.6**	**62.2**	**66.8**	**60.4**	**54.4**	**48.9**	**11.6**	**17.5**	**12.1**	**54.1**	**57.8**	**38.1**	**51.1**	**44.7**	**111.8**

K PCA Rank Sensitivity

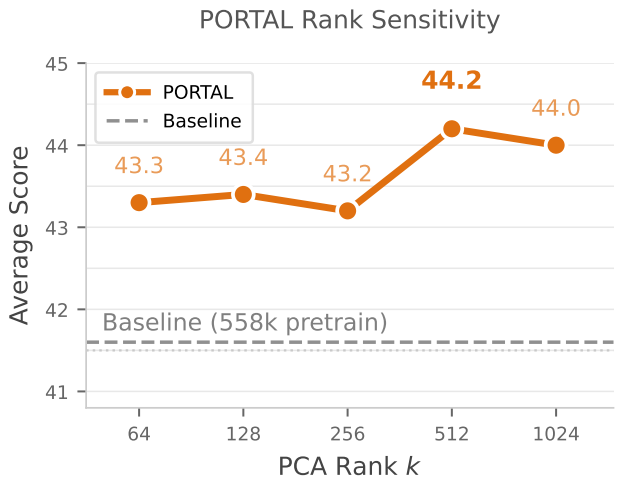


Figure 5: **PCA rank sensitivity** on LLaMA 3.2 1B/CLIP. Average over the 16 benchmarks as a function of the rank k used in the PORTAL construction, with identical Stage-II tuning. All ranks outperform the trained baseline (41.6). k=512 is optimal (44.2).

The PORTAL projector is constructed in a rank-k PCA subspace (Section 4). We vary k over $\{64, 128, 256, 512, 1024\}$ on LLaMA 3.2 1B/CLIP, constructing each projector from the same 1024 COCO images and running identical Stage-II tuning. Figure 5 shows the resulting Avg score. Performance is flat from k=64 to k=256 (approximately 43.3 Avg), peaks at k=512 (44.2), and decreases marginally at k=1024 (44.0). The plateau at low ranks indicates that the dominant PCA directions already capture most of the cross-modal structure. The peak at k=512 suggests that mid-variance directions contribute additional geometric signal. The decrease at k=1024 is consistent with noisy eigenvectors diluting the transport map. We adopt k=512 as the default throughout the paper.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, all claims are placed against related work (Section 2) and all claims regarding the method's performance are backed by empirical results (Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper discusses limitations in the Conclusion and in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, Proposition 1 states its assumptions, and Appendix H.2 gives the complete proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, experimental details are described in the experimental setup (Section 5), and all hyperparameters and the evaluation protocol are given in Appendices B and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, the code is available at https://github.com/mlbio-epfl/PORTAL.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, experimental details are described in the experimental setup (Section 5), and all hyperparameters and the evaluation protocol are given in Appendices B and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, all aggregate scores in the main results and ablation tables are reported with 95% confidence intervals computed from $B{=}1{,}000$ paired bootstrap resamples [61]; resampling is paired across methods within each (encoder, LLM) group so the intervals reflect method differences rather than sample noise. The $\pm$ subscripts in the tables denote the half-width of these CIs. The bootstrap protocol is described in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we specify the computational resources in Appendix B.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we have carefully reviewed and adhered to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, the positive societal impact of the paper is discussed in the introduction and conclusion. Negative societal impacts are not expected as the outlined methodology enables multimodal LLMs projector alignment without paired data while being computationally efficient.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks as beyond code, we are not releasing any new artifacts such as data or models.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we cite the creators of every model and dataset we use (Section 5, Appendices B and C), and Table 5 lists the license of each dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release any new assets beyond code, which is adequately documented in order to reproduce all results from the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper does not involve the development of any LLMs, nor their use beyond minor editing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.