MSP: Modality Self-Play for Training Multimodal LLMs Without Paired Cross-Modal Data

Robert Moseley
University of California, Riverside
rmose009@ucr.edu

Md Kaykobad Reza
University of California, Riverside
mreza025@ucr.edu

Ameya Patil
Amazon
ameyapat@amazon.com

Edward Ayrapetian
Amazon
edayrap@amazon.com

M. Salman Asif
University of California, Riverside
sasif@ucr.edu

Abstract

Large multimodal models have achieved remarkable progress by scaling model capacity and training on massive paired multimodal data. However, this paradigm introduces a fundamental bottleneck: aligned multimodal datasets are expensive to curate, difficult to scale across diverse modality combinations, and increasingly constrain further progress. We introduce *Modality Self-Play* (MSP), a training framework for multimodal foundation models based on proxy-mediated modality composition. Instead of relying on paired cross-modal data, MSP uses anchor-supporting proxy textual descriptions of unseen modalities as semantic bridges during training. Given an observed modality (e.g., image or audio), the model generates proxy descriptions for complementary unseen modalities, which are then encoded and aligned with the observed modality through learned bridges, enabling cross-modal interaction without real paired non-text supervision. Our central hypothesis is that proxy text, combined with contrastively trained pretrained encoders, provides sufficient semantic structure for multimodal composition to emerge. We show that MSP enables zero-shot modality composition: models trained only with a real anchor modality and accompanying proxy descriptions can integrate and reason over multiple modalities at inference time despite never observing real paired non-text data during training. Empirically, MSP achieves state-of-the-art or competitive performance on multiple multimodal benchmarks. Ablation studies further demonstrate that learned bridges improve alignment, while anchor-supporting proxy descriptions enable effective cross-modal composition. MSP removes the need for real paired non-text (e.g., audio-vision) supervision, while continuing to rely on modality-text pairs, pretrained contrastive encoders, and modality-distribution bridge calibration.

1 Introduction

Large multimodal models (LMMs) [11, 44] have advanced the frontier of intelligent systems by enabling unified reasoning over language, vision, audio, and other heterogeneous modalities. Much of this progress has been driven by scaling: larger models, broader modality coverage, and increasingly

Project page: https://csiplab.github.io/Modality-Self-Play/.

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

large paired multimodal datasets have led to strong gains across perception and reasoning tasks [35, 22]. However, most existing approaches rely on a common assumption: effective cross-modal reasoning requires explicit supervision from paired multimodal data. This dependence on paired multimodal data presents a key scalability challenge. High-quality aligned multimodal datasets are expensive to collect, difficult to scale across modality combinations, and often domain-specific. As models expand to incorporate more modalities, the cost of acquiring comprehensive paired data grows combinatorially. As a result, progress in multimodal learning is increasingly constrained by the availability of paired multimodal datasets rather than model design.

In contrast, large amounts of bimodal data (e.g., vision-language or audio-language) are readily available on open source repositories. These datasets contain rich semantic information, raising an important question: *can models learn to reason over multiple modalities without ever observing real paired non-text data during training?* Prior work has explored related directions through shared encoders [9], modular model design [39, 10], and weak supervision [34]. However, most approaches still rely on some form of paired multimodal supervision during training [45] or exhibit limited performance on standard multimodal benchmarks [53, 9, 34].

In this work, we explore an alternative approach based on *proxy-mediated modality alignment*. We introduce **Modality Self-Play (MSP)**, a training paradigm that replaces real paired non-text supervision with proxy textual descriptions. Given an observed modality (e.g., an image), MSP generates proxy text describing complementary unseen modalities (e.g., audio). These proxy descriptions are encoded using contrastively pretrained text encoders (e.g., CLIP/CLAP-style encoders [37, 46]) and projected to the shared language embedding space via a bridge and MLP projector. During training, the model never observes real paired non-text data; instead, it learns from single modalities paired with proxy descriptions that act as semantic bridges. This design leverages the fact that contrastively pretrained encoders provide transferable cross-modal structure [7]. Proxy text does not replicate unseen modalities, but provides a shared semantic space through which representations can be aligned. We therefore do not assume the absence of cross-modal priors, but rather eliminate the need for real paired non-text data during training.

Our central hypothesis is that proxy-mediated alignment is sufficient for multimodal composition to emerge, enabling LMMs to reason jointly over multiple modalities without real paired non-text data during training. We evaluate this hypothesis empirically across multiple multimodal reasoning benchmarks. We show that MSP enables *zero-shot modality composition*: models trained using single modalities and proxy descriptions can reason over multiple modalities at inference time despite never observing them jointly during training. MSP achieves *competitive or state-of-the-art performance* on standard multimodal benchmarks [49, 16, 24] without using real paired non-text training data (Section 4.2). We further analyze the role of proxy descriptions and alignment mechanisms through ablation studies, showing that learned bridges improve alignment between proxy embeddings and the corresponding modality representations, resulting in improved multimodal reasoning performance (Section 4.3). **Our contributions are summarized as follows:**

- We introduce **Modality Self-Play (MSP)**, a training framework that replaces real paired non-text supervision with proxy-mediated alignment.
- We show that proxy textual descriptions can serve as semantic bridges, enabling zero-shot multimodal composition without real paired non-text training data. MSP achieves competitive or state-of-the-art performance on multiple multimodal benchmarks (Section 4.2).
- Ablation studies show that bridges improve alignment between proxy and modality representations, leading to stronger multimodal reasoning performance without real paired non-text data (Section 4.3).

MSP removes the need for real paired non-text (e.g., audio-vision) supervision, while continuing to rely on modality-text pairs, pretrained contrastive encoders, and modality-distribution bridge calibration.

2 Related Work

Our work relates to three lines of research in multimodal foundation models: (i) multimodal LLMs (MLLMs) trained with paired multimodal data, (ii) approaches that align independently observed

modalities to a shared language embedding space, and (iii) text-mediated proxy methods that use language as an intermediate representation.

Training MLLMs with paired multimodal data. The dominant paradigm for MLLMs relies on training with explicitly paired multimodal data, where language or instruction prompts are aligned with one or more observed modalities. Early vision-language models such as BLIP-2 [17] and LLaVA [28] demonstrate strong performance through large-scale image-text pretraining and instruction fine-tuning. This paradigm has been extended to broader modality coverage: NExT-GPT [45], AnyGPT [51], Unified-IO 2 [29], and MIO [43] integrate multiple modalities within unified architectures, while recent omni-modal systems such as Qwen2.5-Omni [35], Baichuan-Omni-1.5 [22], and MiniCPM [50] scale this approach with large multimodal training pipelines. While effective, these methods depend on the availability of aligned paired multimodal datasets, which can be costly to curate and difficult to scale across modality combinations. In contrast, our work explores whether multimodal composition can be learned without requiring explicit paired multimodal supervision.

Multimodal learning via modality-text alignment. A growing line of work reduces reliance on direct multimodal pairing by aligning each modality to language or to a shared embedding space. ImageBind-LLM [10] leverages pretrained modality-specific encoders to align multiple modalities to a shared image embedding space before connecting them to a large language model. PandaGPT [39], X-InstructBLIP [34], and OneLLM [9] extend this approach by aligning multiple modalities to a shared language embedding space through projectors and modality-specific adapters. ChatBridge [53] utilizes language as an intermediate representation connecting modalities, reducing the need for direct modality-to-modality pairing. These approaches demonstrate that language can serve as a unifying interface for multimodal learning. However, they often struggle with cross-modal composition at inference time, since the models are not explicitly trained to reason jointly over multiple modalities. In contrast, MSP introduces proxy-mediated alignment that incorporates compositional signals during training without requiring real paired non-text datasets.

Text-mediated proxies and representation bridging. Several methods use text representations as proxies for other modalities. DeCap [18] and CapDec [33] train decoders using only text embeddings and map visual representations into the text embedding space during inference, enabling zero-shot captioning. Similar ideas have been explored for image, video, and audio captioning [42, 3, 21]. These approaches demonstrate that text can act as an intermediate representation when multimodal data is limited.

However, these methods primarily focus on single-modality generation tasks and do not investigate whether text proxies can be used to learn multimodal composition without paired multimodal data. MSP differs from existing approaches in two important ways. First, proxy text conditioned on an observed modality is generated and used during training to guide cross-modal alignment, rather than serving as a substitute for unseen modalities at inference time. Second, these proxies are used to train the model for cross-modal reasoning by using them as semantic surrogates for unseen modalities within the LLM. This positions MSP as a training-time framework for inducing multimodal composition through proxy-mediated alignment.

3 Method

We introduce *Modality Self-Play* (MSP), a training framework that enables MLLMs to reason jointly over audio and vision at inference time without observing any real audio-vision pairs during training. We define *paired cross-modal data* as samples containing two or more non-text modalities originating from the same underlying source (e.g., audio and vision from the same recording). MSP uses standard modality-text supervision (e.g., image-language or audio-language) to train modality-specific projectors and calibrate the alignment bridges described below, but never uses paired non-text modality data. We first describe the model architecture in Section 3.1, followed by the two training stages: per-modality projector training (Section 3.2) and self-play composition training (Section 3.3).

3.1 Model Architecture

Figure 1 summarizes the full training pipeline, and Figure 2 illustrates the four main components of MSP: modality-specific encoders, modality-specific projectors, text-to-modality bridges, and the language model.

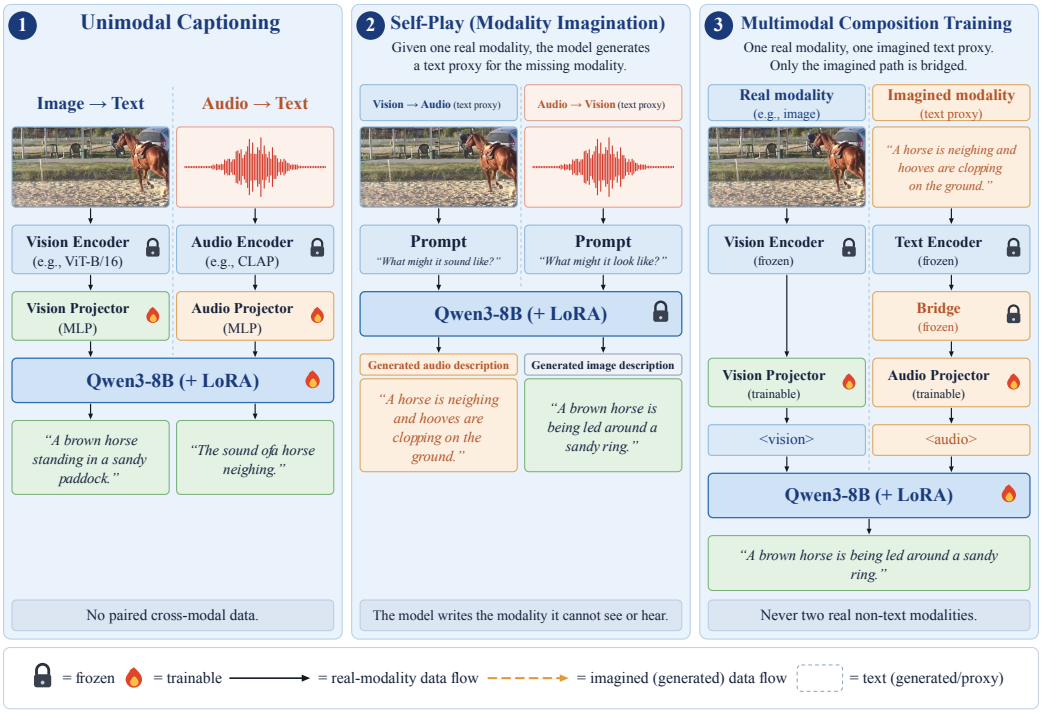

Figure 1: MSP training pipeline. **(1)** Phase 1 trains one projector per modality using modality–text pairs only. **(2)** The frozen Phase-1 model then generates a text proxy for the modality it does not observe. **(3)** Phase 2 trains on one real modality together with the proxy for the missing one. No training sample ever contains two real non-text modalities. A component-level view of the same architecture, with the bridge formulation and the full token sequence, is given in Figure 2.

Modality-specific encoders. We use frozen contrastive encoders to map each non-text modality into a fixed-dimensional feature representation. Specifically, we use LAION-CLAP [46] for audio and MetaCLIP ViT-B/16 [48] for vision. Each encoder produces a single 512-dimensional feature embedding per input. The encoders remain frozen throughout training to ensure that both real and proxy representations reside in a consistent embedding space, enabling the bridge modules to learn meaningful transformations between them.

Modality-specific projectors. Each modality uses a lightweight trainable MLP projector $P_m : \mathbb{R}^{512} \to \mathbb{R}^D$ to map modality-specific features into the language embedding space, where D=4096 is the hidden dimension of Qwen3-8B. Given an input x_m from modality m, the encoder produces a feature representation z_m, which is projected into the language embedding space as

$$h_m = P_m(E_m(x_m)). \tag{1}$$

Each modality has its own projector. We represent every modality using a single token, ensuring that both real and proxy features share the same representation format.

LLM backbone. We use Qwen3-8B [36] as the language backbone. The input prompt contains modality-specific special tokens, `<audio>` and `<vision>`, corresponding to audio and vision inputs, respectively. During the forward pass, these tokens are replaced with the projected modality embeddings h_m, following the standard design used in multimodal LLMs [28, 17]. We apply LoRA adapters to the attention and MLP layers of Qwen3-8B while keeping the remaining LLM parameters frozen.

3.2 Phase 1: Per-Modality Projector Training

In Phase 1, we train the audio and vision projectors independently using standard modality-caption datasets. For each batch from modality $m \in \{a, v\}$, the input prompt contains a single special token (e.g., `<audio>` or `<vision>`) along with the text instruction. The corresponding modality-specific

feature is computed and projected into the LLM embedding space using (1), which replaces the special token in the prompt. The encoder E_m is frozen and used only for feature extraction, while the projector P_m and the LoRA adapters in the LLM backbone are trained. Training is performed using standard next-token prediction over the response tokens:

$$\mathcal{L}_{\mathrm{P1}} = -\sum_{j \in \mathcal{R}} \log p_\theta(y_j \mid y_{<j}, q, h_m), \tag{2}$$

where q denotes the text prompt, $\mathcal{R}$ denotes response tokens, and both prompt and placeholder tokens are excluded from the loss computation. The resulting Phase 1 model serves two roles in Phase 2: it initializes the modality-specific projectors and LoRA adapters, and it is used as the generator for self-play proxy text describing unseen modalities.

3.3 Phase 2: Self-Play Composition Training

Phase 2 enables the model to learn cross-modal composition between audio and vision without using any real paired non-text data. Since no audio-vision pairs are available, each training example uses a single real modality as an *anchor* (e.g., vision) and a *proxy text description* for the unseen modality (e.g., audio). The proxy text is first encoded using a contrastive text encoder and then transformed through a learned text-to-modality bridge before being mapped into the LLM embedding space via the modality-specific projector. This allows proxy representations to be injected into the model through the same interface as real modality features. When preparing our dataset, any data that could be considered "paired" by definition (i.e., containing both vision and audio) was exclusively utilized for one modality for a single training sample. If, for example, there was some sample containing both vision and audio, we designated the sample for either being used with its audio and not with its vision, or vice versa. This means that any given sample would only ever be used with one of its modalities—not one pass with vision, and a later pass with its audio. We refer to this as the one-real-anchor rule.

In the following, we describe the proxy generation process, the text-to-modality bridge that aligns proxy embeddings with real modality representations, and the Phase 2 training objective.

Self-Play Proxy Generation. For each Phase-2 example, we select one modality as the anchor (e.g., audio) and a different modality as the target (e.g., vision). The Phase-1 model is conditioned on the anchor via its corresponding pathway and prompted to generate a short text description $t'_{m'}$ of the unseen modality m'. We use prompt templates aligned with the original training style of the contrastive encoders (AudioCaps-style for CLAP, COCO-style for MetaCLIP); full prompts are provided in Appendix C. Since the proxy is generated from the anchor, it is anchor-supporting by construction. All proxies are generated once before Phase 2 and cached, so no online generation is required during training.

Text-to-Modality Bridge. The text-encoder and modality-encoder outputs of the contrastive encoders lie in related but shifted regions of a shared embedding space [25]. Directly feeding text-encoder features into the projector P_m would lead to a distribution mismatch, since P_m is trained on modality-encoder features during Phase 1. To address this, we learn a linear bridge B_m that maps the embeddings from the text-encoder into the modality-encoder embedding space. The bridge is trained via ridge regression on 1,000 paired (caption, modality) samples (AudioCaps [14] for audio and LLaVA-665K [27] for vision), split into 500 training and 500 held-out samples:

$$W_m = (T^\top T + \lambda I)^{-1} T^\top R, \quad B_m(t) = W_m^\top t, \quad \lambda = 1.0, \tag{3}$$

where T and R denote stacked text-tower and modality-tower embeddings, respectively. The bridge is trained once before Phase 2 and kept fixed thereafter. The resulting bridged proxy feature $\hat{z}_{m'} = B_{m'}(E^{\mathrm{text}}_{m'}(t'_{m'}))$ is then passed through the projector $P_{m'}$ and inserted at the corresponding modality special token using the same mechanism as real modality inputs. Bridge calibration is evaluated in Appendix H, and remains frozen during Phase 2 training.

Phase 2 Forward Pass. Let m be the real anchor modality and m' be the missing target modality. Phase 2 constructs one real feature, $h_m = P_m(E_m(x_m))$, and one bridged proxy feature, $h'_{m'} = P_{m'}(B_{m'}(E^{\mathrm{text}}_{m'}(t'_{m'})))$. Both vectors have the same LLM hidden width D and are inserted into the prompt at their reserved modality placeholders. An audio-anchored example therefore contains real audio at `<audio>` and a bridged vision proxy at `<vision>`; a vision-anchored example reverses the roles. This construction gives real and proxy inputs identical token shape and insertion semantics.

Phase 2 Training and Data. Audio anchors are drawn from Clotho-AQA [26], AudioCaps [14], WavCaps [32], and from the audio tracks of MSRVTT-QA and MSVD-QA training videos [47]. Vision anchors are drawn from VQAv2 [8], LLaVA-665K [27], and additional frames from MSRVTT-QA and MSVD-QA. The trainable parameters are the modality-specific projectors and shared LoRA adapters, while the modality-specific encoders and bridges remain frozen. The objective is the same as (2), now conditioned on both the real anchor and the proxy feature. This enables cross-modal composition during training without requiring any real paired audio–vision data.

Inference. At inference time, real modalities are encoded and projected through the same encoder-projector pathway used during training. Tables 5 and 6 demonstrate that despite the LLM's own ability to easily distinguish between proxies and modalities in residual space, the model converges to the same tokens in the output layer indicating that the model is able to use proxies as substitutes for real modalities during training. We do not generate proxies at inference.

Figure 3 (Appendix G) shows representative proxy examples generated for AVQA and MUSIC-AVQA samples. These examples illustrate the intended role of proxy text: it should preserve the semantic content of the anchor while occupying the representation space of the missing modality.

4 Experiments and Results

4.1 Dataset and Implementation Details

Phase 1 data. In Phase 1, we train the modality-specific projectors together with the LLM decoder using standard modality-text pairs to align each modality into a shared language embedding space that can be processed by the LLM. We represent each modality using a single token. We use the audio-text pairs from AudioCaps [14] and WavCaps [32] datasets, while image-text pairs are obtained from the LLaVA-665K dataset [27]. All datasets and pretrained encoders used in this work are publicly released artifacts, used under the license and terms of use of their original release.

Proxy generation and Phase 2 data. Self-play proxies are generated after Phase 1 and cached for the remainder of training. For each training sample, the Phase 1 model is conditioned on the observed anchor modality and prompted to generate textual descriptions of the unseen target modality in its native captioning style. These proxy descriptions are then used as semantic surrogates for the missing modalities during subsequent MSP training. To generate vision proxies, we use audio samples from the Clotho-AQA [26], AudioCaps [14], and WavCaps [32] datasets, together with the audio tracks of MSRVTT-QA and MSVD-QA [47]. Conversely, to generate audio proxies, we use images from VQAv2 [8] and LLaVA-665K [27], together with frames from MSRVTT-QA and MSVD-QA. For video samples, we choose either audio or uniformly sample 4 frames, encode each with MetaCLIP ViT-B/16, and mean-pool the resulting features into a single 512-dimensional vision token.

Importantly, MSP never observes paired audio-vision-text modalities during composition training. During Phase 2 training, we alternate between *audio + proxy vision description* and *vision + proxy audio description* samples to encourage proxy-mediated multimodal composition.

Bridge calibration. Each bridge is a linear map that re-projects text-tower vectors into the modality-tower space, correcting the offset between a contrastive encoder's two towers [25]. We fit one per modality using the ridge regression and calibration split introduced in Section 3.3 (λ=1.0). Both bridges are fit once before Phase 2 begins and then frozen.

Evaluation benchmarks. We evaluate our method on three cross-modal question answering (QA) benchmarks that require joint reasoning over audio and vision to answer each question: **AVQA** [49], **MUSIC-AVQA** [16], and **OmniBench** [24]. AVQA and MUSIC-AVQA focus on audio-visual question answering, while OmniBench extends this setting to audio-visual-text reasoning, enabling evaluation in a broader multimodal context. Here, "zero-shot" means we do not finetune specifically for performance on these benchmarks (meaning we do not use the training data for these benchmarks to enhance performance).

Implementation details. We use Qwen3-8B [36] with LoRA adapters applied to attention and MLP modules with ranks $r \in \{128, 256, 512\}$. The modality-specific projectors are MLPs and are fully trainable. Frozen contrastive encoders: MetaCLIP B/16 [48] for vision and LAION-CLAP [46] for audio; per-modality features are pooled to a single 512-dim token before projection. Full optimization

Table 1: Performance comparison across different multimodal models on the MUSIC-AVQA test split. Primarily, we wanted to compare against models that were allowed to use paired data but did not specifically finetune for performance on the listed benchmarks.

Method	Parameters	Performance
ImageBind-LLM [10]	7B	39.72
X-InstructBLIP [34]	7B	28.10
X-InstructBLIP [34]	13B	44.50
ChatBridge [53]	13B	43.00
OneLLM [9]	7B	47.60
Qwen2.5-VL [1]	7B	47.75
GPT-4o [12]	-	50.08
MSP (LoRA r = 128)	8B	**60.24**
MSP (LoRA r = 256)	8B	**64.17**
MSP (LoRA r = 512)	8B	**65.65**

Table 2: Performance comparison across different multimodal models on the AVQA test split. Primarily, we wanted to compare against models that were allowed to use paired data but did not specifically finetune for performance on the listed benchmarks.

Method	Parameters	Performance
PSAC [20]	-	78.60
ACRTransformer [52]	-	81.70
HME [4]	-	81.80
LADNet [19]	-	81.90
SSAM [38]	7B	82.12
HGA [13]	-	82.20
HCRN [15]	-	82.50
ImageBind-LLM [10]	7B	55.00
X-InstructBLIP [34]	7B	44.29
MSP (LoRA r = 128)	8B	**85.70**
MSP (LoRA r = 256)	8B	**85.73**
MSP (LoRA r = 512)	8B	**85.58**

details, including the projector-warmup schedule and per-rank effective batch sizes, are listed in Appendix D.

4.2 Performance on Multimodal Benchmarks

We evaluate MSP on three multimodal benchmark datasets and compare it against state-of-the-art multimodal models. We report results for models trained with LoRA ranks $\{128, 256, 512\}$. Despite never observing real paired non-text data during training, MSP achieves competitive performance and, in several cases, outperforms baseline MLLMs on multimodal question answering tasks that require joint reasoning over paired modalities. We report audio-vision performance for all the datasets.

Performance on MUSIC-AVQA benchmark. Table 1 reports zero-shot accuracy on the MUSIC-AVQA benchmark. MSP with LoRA rank $r{=}512$ achieves 65.65, outperforming OneLLM by $+18.05$ pp and GPT-4o by $+15.57$ pp, despite GPT-4o having a substantially larger parameter count. The $r{=}256$ and $r{=}128$ configurations achieve 64.17 and 60.24, respectively, both surpassing the baseline methods. Unlike OneLLM, which is trained only on modality-text pairs, MSP additionally uses proxy textual descriptions of unseen modalities during training. This enables effective reasoning over paired audio-visual inputs at inference time and demonstrates the effectiveness of proxy supervision for learning cross-modal composition.

Performance on AVQA. Table 2 reports performance on the AVQA benchmark, where MSP achieves 85.73 in zero-shot setting with LoRA rank $r{=}256$, representing the highest reported zero-shot AVQA performance among methods not finetuned on the AVQA dataset. Performance remains consistent across ranks $r{\in}\{128, 256, 512\}$, suggesting performance saturation rather than insufficient model

Table 3: Performance comparison across different multimodal models on OmniBench test split. Primarily, we wanted to compare against models that were allowed to use paired data but did not specifically finetune for performance on the listed benchmarks.

Method	Parameters	Performance
AnyGPT [51]	7B	18.04
UnifiedIO2-large [29]	1.1B	27.06
Baichuan-Omni [23]	7B	32.20
VITA [5]	8×7B	33.10
VITA-1.5 [6]	7B	33.40
UnifiedIO2-xxlarge [29]	6.8B	33.98
video-SALMONN [40]	13B	35.64
OpenOmni [30]	7B	37.40
UnifiedIO2-xlarge [29]	3.2B	38.00
MiniCPM-o 2.6 [50]	7B	40.50
Baichuan-Omni-1.5 [22]	7B	42.90
Gemini-1.5-Pro [41]	-	42.91
Qwen2.5-Omni [35]	7B	**56.13**
MSP (LoRA r = 128)	8B	42.12
MSP (LoRA r = 256)	8B	**43.26**
MSP (LoRA r = 512)	8B	41.07

Table 4: Proxy and bridge help improve model performance. Every cell is at r=512 on the test splits.

Method	AVQA			MUSIC-AVQA		
	Audio	Video	Audio-Video	Audio	Video	Audio-Video
Without proxy and bridge	59.55	63.91	70.21	44.03	44.64	45.75
With proxy without bridge	78.53	79.50	82.91	45.85	54.49	58.60
With proxy and bridge	**80.24**	**81.33**	**85.58**	**57.26**	**57.11**	**65.65**

capacity. MSP outperforms zero-shot baselines such as ImageBind-LLM and X-InstructBLIP by large margins (30.73–41.44 pp). Notably, MSP also surpasses methods including PSAC, HGA, and HCRN, despite those models being trained on the paired AVQA train set. These results highlight the effectiveness of MSP in learning cross-modal composition without real paired non-text training data.

Performance on OmniBench. Table 3 reports performance on the OmniBench benchmark, which requires joint reasoning over vision, audio, and text modalities. MSP with LoRA rank r=256 achieves 43.26, outperforming most baseline methods and achieving performance competitive with Baichuan-Omni-1.5 and Gemini-1.5-Pro despite never being trained on real paired non-text data. Notably, the closed-source Gemini-1.5-Pro achieves 42.91, slightly below MSP on this benchmark. MSP is the only method in this comparison trained without real paired cross-modal data, and it remains competitive with models that use it, further demonstrating the effectiveness of proxy-based cross-modal composition.

4.3 Learning Modality Composition via Proxy

We analyze the model to understand how proxy supervision and the text-to-modality bridge contribute to learning cross-modal composition. Table 4 compares no proxies and no bridge, proxies without a bridge, and full MSP with both proxies and bridge. At the audio-video setting, proxies are worth +12.70 on AVQA and +12.85 on MUSIC-AVQA, and the bridge a further +2.67 and +7.05. This shows that proxies provide strong learning signals for modality composition, while the bridge further improves alignment.

We also examine representation behavior across LLM layers in Table 5 using both residual-stream similarity and the *logit lens*. While residual similarity decreases across layers, vocabulary-space similarity recovers at the final layer and ends well above it ($\Delta = +0.166$ audio, $+0.124$ vision). This indicates that proxies do not need to match real modality features in hidden space; instead, the model learns to map them to similar output predictions, enabling effective cross-modal composition.

Table 5: Paired residual and output-layer cosine similarity. Each proxy is matched to its nearest real counterpart by last-layer residual cosine, and similarities are computed over those pairs. Here, Δ = vocab $\cos_{L_{\text{last}}}$ − residual $\cos_{L_{\text{last}}}$. These are not comparable to the trajectory cosines in Table 10, which measure a different quantity.

Modality	residual cos (L_0/L_{mid}/L_{last})	vocab cos (L_0/L_{mid}/L_{last})	Δ at last layer
Audio	.956 / .889 / .778	.953 / .913 / .944	$+\mathbf{0.166}$
Vision	.821 / .883 / .805	.828 / .799 / .929	$+\mathbf{0.124}$

We fit a logistic-regression probe at the input, middle, and final layers of the LLM backbone to predict real vs. bridged-proxy from the hidden state at that layer (n=400 per cell). Real and bridged-proxy features remain linearly separable at every layer for both audio and vision (Table 6); MSP does not erase the distinction in the residual stream. This supports the interpretation that the LM-head readout, rather than residual indistinguishability, explains the proxy–real alignment.

Table 6: Real and proxy hidden states remain distinguishable. A logistic-regression probe predicts whether a hidden state came from a real feature or a bridged-proxy feature at three layers; n=400 per cell. Values are probe accuracies, so higher values mean real and proxy states are easier to separate. The high accuracies show that MSP does not make the hidden states identical; Table 5 tests whether the LM-head readout is nevertheless aligned.

Modality	L_0	L_{mid}	L_{last}
Audio	0.953	0.970	0.963
Vision	1.000	1.000	0.998

Additional alignment analyses are provided in Appendix E.

Phase 2 trains on QA-formatted data, so part of the gain over Phase 1 comes from that supervision alone rather than from composition. Separating the two (Table 13), we still find that MSP composition training with proxies enables significantly better performance than the QA supervision by itself.

Real modalities vs. proxy modalities at inference. We found across the board that real modalities outperform their proxy counterparts (Table 7a). Token count, position, attention mask, and decoding are unchanged. Proxies for this ablation are generated via the same mechanism used during the training of the model.

Token-matched controls. Table 7b varies the audio slot with real vision and the vision slot with real audio. The key takeaway is that the model utilizes real modality content through these slots, and that only correct content recovers full performance. Additional details are provided in Appendix I. Question-type breakdowns at r=512 are provided in Appendix B.

Sensitivity to proxy quality. As might be expected based on the token-matched controls, the quality and content of the modality given to the model has a large impact. While we do not have results for a model trained on degenerate proxies, the token-matched controls indicate that the model is reading real modality content from the token slot, thus generating realistic proxies is also important for the downstream model to learn the desired cross-modal relationships (Appendix K).

MUSIC-AVQA-R results are provided in Appendix J.

5 Limitations and Future Directions

While MSP demonstrates that proxy-mediated alignment can enable multimodal composition without real paired non-text supervision, it relies on the quality and semantic consistency of proxy textual descriptions. Noisy or weakly aligned proxies can degrade model performance (Appendix K). Additionally, MSP builds on contrastively pretrained encoders (e.g., CLIP/CLAP-style models) to provide cross-modal structure, and therefore inherits their coverage limitations and potential biases. Future work can address these limitations along several directions. Improving proxy generation—through stronger generative models or retrieval-based methods—could yield more accurate and semantically

Table 7: Real modalities vs. proxy modalities at inference and token-matched controls. Qwen3-8B LoRA r=512, test splits: AVQA (n=16,910), MUSIC-AVQA (n=9,185).

(a) Real modalities vs. proxy modalities at inference

Audio	Vision	AVQA	MUSIC-AVQA
real	real	85.58	65.65
imagined	real	81.19	57.34
real	imagined	79.28	62.87
imagined	imagined	75.75	41.33

(b) Token-matched controls

	Audio slot		Vision slot	
	AVQA	MUSIC-AVQA	AVQA	MUSIC-AVQA
Real	85.58	65.65	85.58	65.65
Learned null	81.50	60.48	80.24	61.35
Zero	80.28	56.76	78.35	59.01
Random (moment-matched Gaussian)	79.66	56.08	74.97	58.08
Shuffled (real feature, wrong example)	76.81	52.95	72.89	54.48

grounded descriptions. Reducing the training–inference gap, either by better integrating proxy information at inference or by learning proxy-free alignment mechanisms, is another promising avenue. Extending MSP to a wider range of modalities and more complex reasoning tasks, as well as studying its behavior at larger scales, would further evaluate its generality. Finally, developing a deeper theoretical understanding may help clarify when and why multimodal composition can emerge without real paired non-text supervision. Future work should focus on using MSP in paradigms with significantly more tokens and also determine how closely MSP compares with paired data ceilings (or if they can be exceeded by using additional text proxies; Appendix L).

Broader impacts. This work improves multimodal reasoning without requiring real paired non-text data, which can benefit data-limited or privacy-sensitive applications. However, improved cross-modal synthesis may also facilitate misuse such as multimodal deepfakes or fabricated content. To mitigate these dual-use risks, we advocate for the use of ethically sourced training data, model access control, and the adoption of provenance-tracking techniques for any synthesized outputs.

6 Conclusion

In this work, we revisited a central assumption in multimodal learning that large-scale paired multimodal data is necessary for learning cross-modal reasoning. We introduced **Modality Self-Play (MSP)**, a training paradigm in which proxy textual descriptions serve as semantic bridges between observed and unseen modalities. MSP removes the need for real paired non-text (e.g., audio-vision) supervision, while continuing to rely on modality-text pairs, pretrained contrastive encoders, and modality-distribution bridge calibration. Our results show that proxy-based supervision is both viable and effective. Models trained with MSP demonstrate strong performance on standard multimodal benchmarks, while reducing reliance on costly data curation pipelines. More broadly, our findings suggest that multimodal composition can emerge from structured interactions over independently learned representations mediated through proxy text, rather than from exhaustive aligned multimodal data. We hope this work encourages the development of more scalable and data-efficient approaches to multimodal learning based on principled alignment mechanisms.

Acknowledgments, funding, and competing interests

This work is partially supported by a gift award from Amazon and an NSF award 2406199. This work used Anvil at Purdue University through allocation ELE260028 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [2], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. Ameya Patil and Edward Ayrapetian are employed by Amazon. The authors declare no other competing interests.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing (PEARC '23)*, pages 173–176. ACM, 2023.

[3] Soham Deshmukh, Benjamin Elizalde, Dimitra Emmanouilidou, Bhiksha Raj, Rita Singh, and Huaming Wang. Training audio captioning models without audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[4] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1999–2007, 2019.

[5] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Yuhang Dai, Meng Zhao, Yi-Fan Zhang, Shaoqi Dong, Yangze Li, Xiong Wang, et al. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*, 2024.

[6] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, et al. VITA-1.5: Towards GPT-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025.

[7] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913, 2017.

[9] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. Onellm: One framework to align all modalities with language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26584–26595, 2024.

[10] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*, 2023.

[11] Dawei Huang, Chuan Yan, Qing Li, and Xiaojiang Peng. From large language models to large multimodal models: A literature review. *Applied Sciences*, 14(12):5068, 2024.

[12] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[13] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11109–11116, 2020.

[14] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019.

[15] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9972–9981, 2020.

[16] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] 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 *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[18] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. DeCap: Decoding CLIP latents for zero-shot captioning via text-only training. In *International Conference on Learning Representations (ICLR)*, 2023.

[19] Xiangpeng Li, Lianli Gao, Xuanhan Wang, Wu Liu, Xing Xu, Heng Tao Shen, and Jingkuan Song. Learnable aggregating net with diversity learning for video question answering. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1166–1174, 2019.

[20] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8658–8665, 2019.

[21] Xiquan Li, Wenxi Chen, Ziyang Ma, Xuenan Xu, Yuzhe Liang, Zhisheng Zheng, Qiuqiang Kong, and Xie Chen. DRCap: Decoding CLAP latents with retrieval-augmented generation for zero-shot audio captioning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[22] Yadong Li, Jun Liu, Tao Zhang, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, et al. Baichuan-Omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*, 2025.

[23] Yadong Li, Haoze Sun, Mingan Lin, Tianpeng Li, Guosheng Dong, Tao Zhang, Bowen Ding, Wei Song, Zhenglin Cheng, Yuqi Huo, et al. Baichuan-Omni technical report. *arXiv preprint arXiv:2410.08565*, 2024.

[24] Yizhi Li, Yinghao Ma, Ge Zhang, Ruibin Yuan, Kang Zhu, Hangyu Guo, Yiming Liang, Jiaheng Liu, Zekun Wang, Jian Yang, et al. Omnibench: Towards the future of universal omni-language models. *arXiv preprint arXiv:2409.15272*, 2024.

[25] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[26] Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. Clotho-AQA: A crowdsourced dataset for audio question answering. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1140–1144. IEEE, 2022.

[27] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, 2024.

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

[29] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024.

[30] Run Luo, Ting-En Lin, Haonan Zhang, Yuchuan Wu, Xiong Liu, Min Yang, and Yongbin Li. OpenOmni: Advancing open-source omnimodal large language models with progressive multimodal alignment and real-time self-aware emotional speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[31] Jie Ma, Min Hu, Pinghui Wang, Wangchun Sun, Lingyun Song, Hongbin Pei, Jun Liu, and Youtian Du. Look, listen, and answer: Overcoming biases for audio-visual question answering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[32] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3339–3354, 2024.

[33] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected CLIP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

[34] Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv preprint arXiv:2311.18799*, 2023.

[35] Qwen Team. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

[36] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[38] Md Kaykobad Reza, Ameya Patil, Edward Ayrapetian, and M. Salman Asif. Ssam: Singular subspace alignment for merging multimodal large language models. *arXiv preprint arXiv:2603.21584*, 2026.

[39] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*, pages 11–23, 2023.

[40] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. video-SALMONN: Speech-enhanced audio-visual large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[41] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[42] Junyang Wang, Ming Yan, Yi Zhang, and Jitao Sang. From association to generation: Text-only captioning by unsupervised cross-modal mapping. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4326–4334, 2023.

[43] Zekun Wang, King Zhu, Chunpu Xu, Wangchunshu Zhou, Jiaheng Liu, Yibo Zhang, Jiashuo Wang, Ning Shi, Siyu Li, Yizhi Li, et al. MIO: A foundation model on multimodal tokens. *arXiv preprint arXiv:2409.17692*, 2024.

[44] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023.

[45] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. In *Forty-first International Conference on Machine Learning*, 2024.

[46] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[47] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM International Conference on Multimedia (MM)*, pages 1645–1653, 2017.

[48] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *International Conference on Learning Representations (ICLR)*, 2024.

[49] Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. AVQA: A dataset for audio-visual question answering on videos. In *Proceedings of the 30th ACM International Conference on Multimedia (MM)*, 2022.

[50] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[51] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9637–9662, 2024.

[52] Jipeng Zhang, Jie Shao, Rui Cao, Lianli Gao, Xing Xu, and Heng Tao Shen. Action-centric relation transformer network for video question answering. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):63–74, 2020.

[53] Zijia Zhao, Longteng Guo, Tongtian Yue, Sihan Chen, Shuai Shao, Xinxin Zhu, Zehuan Yuan, and Jing Liu. ChatBridge: Bridging modalities with large language model as a language catalyst. *arXiv preprint arXiv:2305.16103*, 2023.

A Detailed architecture diagram

Figure 2 gives the component-level view of MSP referred to in Section 3.1. It shows the frozen contrastive encoders and their text towers, the trainable per-modality projectors, the frozen text-to-modality bridge used on the proxy path in Phase 2, the reserved token slots in the prompt, and the Qwen3-8B backbone with its LoRA adapters.

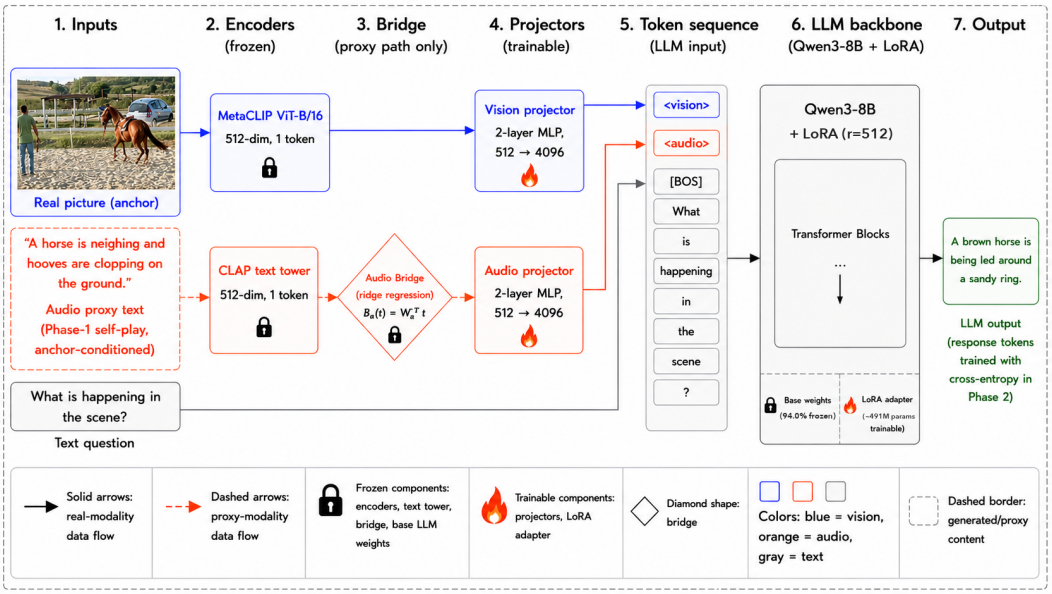

Figure 2: MSP architecture, component-level view. MSP uses frozen audio and vision encoders with their associated text encoders, trainable per-modality projectors, a frozen text-to-modality bridge for Phase 2, and a Qwen3-8B language model with LoRA adapters.

B Per-question-type breakdowns

The single-token model does not collapse: AVQA *Come From* (sound localization) 90.9 and *Where* (spatial) 80.3; MUSIC-AVQA *Counting* 72.6 and *Location* 57.2 (recomputed using the r=512 checkpoint; Table 8).

Table 8: Reported question-type and overall modality results at r=512. The question-type block reports the four categories recomputed for this checkpoint.

Dataset	Question type	Accuracy
AVQA	Come From (sound localization)	90.9
AVQA	Where (spatial)	80.3
MUSIC-AVQA	Counting	72.6
MUSIC-AVQA	Location	57.2

Dataset	Audio	Video	Audio-Video
AVQA	80.24	81.33	85.58
MUSIC-AVQA	57.26	57.11	65.65

C Self-play prompt templates

Phase 2 proxies are generated by the Phase-1 checkpoint conditioned on the real anchor modality. The prompt template controls which target encoder distribution the generated text is meant to match;

we use encoder-native terse prompts that mirror the caption style each contrastive encoder was trained on (Appendix H). The verbatim templates used throughout this paper are reproduced below.

Audio proxy from a vision anchor (CLAP-targeting).

> `<vision>`
>
> Describe the sounds in this scene as a single AudioCaps-style caption: ONE short sentence, 8–15 words, present tense, naming 1–3 specific sound sources directly. Do not use multiple sentences and do not describe pitch, timbre, or volume.
>
> Do NOT use: image, picture, photo, visible, scene, look, see, shown, view, visually.
>
> Examples:
> "A man is speaking while water is flowing."
> "A train engine is running and blowing its horn."
> "Music is playing as a vehicle passes by."
> "An animal is chewing loudly in the background."
> "Wind is blowing and leaves are rustling."
>
> Write ONE such sentence. No other text.

Vision proxy from an audio anchor (MetaCLIP-targeting).

> `<audio>`
>
> Write a single COCO-style image caption describing what the source of the sound would look like as a photograph. Use one short sentence, 8–18 words, present tense.
>
> Do NOT use any of these words or concepts: sound, audio, noise, hear, hearing, listen, volume, pitch, timbre, loud, quiet, rhythmic, echo.
>
> Name the main object, person, or animal and the setting. Mention visible attributes only.
>
> Examples:
> "A dog stands on grass with its mouth open."
> "A man sits indoors beside a running kitchen sink."
> "A train passes along tracks near a platform."
>
> Write ONE such sentence. No other text.

Both templates share a common structure: one paragraph naming the target encoder's caption-style family, one paragraph of negative-word constraints to prevent leakage from the source modality's vocabulary, and a short positive instruction listing the content axes the target caption distribution covers.

D Optimization

Both Phase 1 and Phase 2 use AdamW with bfloat16 mixed precision, peak learning rate 5×10^{-5}, cosine decay, one epoch per phase, and seed 42. Effective batch size is 16 (micro-batch 2 $\times$ gradient-accumulation 8) for LoRA $r{=}128$ and 16 (micro-batch 1 $\times$ gradient-accumulation 16) for $r \in \{256, 512\}$. Each reported LoRA run uses DeepSpeed ZeRO-2 on one NVIDIA H100 (80 GB). Training and evaluation for the reported models consumed approximately $1{,}000$ H100 GPU-hours in total.

Phase 1 schedule. For the first $2{,}000$ optimizer steps the LLM is frozen and only the modality projectors are updated; afterward the LoRA parameters are unfrozen. Each Phase 1 batch carries real features from exactly one modality, and audio and vision batches alternate so the update parameters see both pathways within each rotation. The Phase-1 checkpoint is then used both as the initialization for Phase 2 and as the generator that produces the cached self-play proxies.

Phase 2 schedule. Phase 2 continues from the Phase-1 checkpoint with frozen bridges and the cached self-play proxies; only the modality projectors and the chosen LLM update parameters are trained. Dataset sources for both phases are listed in Section 4.1.

One additional note is that for all models, modality dropout is applied during training so that absent modalities are not treated as malformed input.

E Detailed mechanism probes

To analyze the effect of bridge on modality alignment, we calculate the cosine similarity between proxy and real modality features. As shown in Table 9, alignment improves substantially after applying the bridge, increasing similarity from 0.218–0.422 in the raw encoder space to 0.548–0.823 after bridging (a $1.8\times$–$2.6\times$ increase), with additional smaller gains from the projector.

Table 9: Bridge and projector progressively align proxy with real-modality features.

Dataset	**Modality**	**Pre-bridge** (encoder only)	**Pre-projector** (encoder+bridge)	**Post-projector** (encoder+bridge+projector)
AVQA	Vision	0.307	0.786 **(+0.479)**	**0.889** **(+0.582)**
	Audio	0.218	0.548 **(+0.330)**	**0.697** **(+0.479)**
MUSIC-AVQA	Vision	0.333	0.823 **(+0.490)**	**0.893** **(+0.560)**
	Audio	0.422	0.752 **(+0.330)**	**0.821** **(+0.399)**

Table 10 reports the companion surrogate-input control used in Section 4.3. Zero and Gaussian features reduce placeholder engagement, while valid modality features remain closer to the real-feature trajectory. The cosines in this table are trajectory similarities against the unmodified real-input run on the same example, whereas Table 5 reports similarity between nearest-matched proxy–real pairs. The two quantities are measured differently and their magnitudes are not comparable.

Table 10: Valid features engage the placeholder. We compare real features with surrogate inputs using the last-layer residual cosine of the whole hidden-state trajectory against the unmodified real-input run on the same example (n=200 pairs). This is a different quantity from the nearest-pair cosine in Table 5 and the two should not be compared directly. Higher is closer to the real trajectory. Zero and Gaussian inputs reduce engagement; valid features remain closer.

Modality	bridged (ours)	zero	Gaussian	wrong-mod	shuffled-real
audio	0.291	**0.097**	**0.136**	0.309	0.316
vision	0.321	**0.208**	**0.237**	0.391	0.432

F Encoder selection

Our initial hypothesis when selecting encoders for our final model was that whichever encoders had the highest cosine similarity between text proxies passing through their text encoder and their modality counterparts would be the most useful for MSP. We calculated the mean cosine similarity between all COCO images and their text captions (Table 11). Cosine similarity is not the only determinant of

Table 11: Mean cosine similarity between all COCO images and their text captions, used to inform encoder selection.

Vision encoder	Mean cosine similarity
MetaCLIP B/16	0.407
OpenAI CLIP B/16	0.347
DFN2B B/16	0.331
SigLIP2 B/16	0.152
SigLIP1 B/16	0.143

success with a given contrastive encoder, but we believe that it does matter for effectiveness of MSP training. One note is that sigmoid-loss models produce lower raw cosine values by design (SigLIP). For our final model that we report in the paper we utilized MetaCLIP due to its strong text-vision alignment and highest starting cosine similarity. This process was also repeated with a selection of audio encoders to inform our final model architecture.

G Anchor-supporting proxy examples

(a) MUSIC-AVQA – *audio anchor → vision proxy*

Image	Real anchor (audio)	Generated proxy (vision)	Held-out GT
	*Audio clip of a ukulele and guitar playing simultaneously.*	"A red acoustic guitar and blue ukulele on a wooden stage. The guitarist strums chords while the ukulele player plucks strings."	Two musicians on stage; acoustic guitar and ukulele playing together.

(b) AVQA – *vision anchor → audio proxy*

Image	Real anchor (vision)	Generated proxy (audio)	Held-out GT
	*Image of a brown horse in a sandy ring with a person leading it.*	"A horse is neighing and hooves are clopping on the ground."	Horse neighing and hooves clopping on hard ground (audio held out).

Figure 3: Examples of anchor-supporting proxy generation. Each panel shows a real anchor in one modality and the proxy text generated for the other modality. The held-out ground truth is shown only for context; it is not used to generate the proxy.

The qualitative figure (Figure 3) shows faithful proxy generations, where the Phase-1 model correctly describes the held-out modality. The two examples below (Figure 4) illustrate a more nuanced case: the proxy is not literally faithful to the held-out scene, but remains consistent with the real anchor. MSP can still benefit from these examples because the training signal is anchor consistency rather than literal reconstruction of the held-out modality, matching the shuffled-input result in Section 4.3.

(a) AVQA, sample `NumXu4eOqAw_000006` — white double-decker bus



Real audio anchor: engine humming with tires rolling over pavement.
Generated vision proxy: "A silver car speeds down a curvy mountain road at sunrise."
Mismatch: the real scene is a white double-decker bus on a highway; the proxy invents a silver car on a mountain road. Both descriptions are consistent with the engine + tires audio.

(b) AVQA, sample `xeAxybX3i1E_000030` — waterfowl



Real audio anchor: ducks calling near water.
Generated vision proxy: "A dog grasping a duck."
Mismatch: the real scene shows ducks in water; the proxy describes a different scene that shares the duck object but introduces a dog. The audio (duck calls) remains consistent with both interpretations.

Figure 4: Anchor-supporting proxy examples. Generated vision proxies can differ from the held-out scene while remaining compatible with the real audio anchor.

H Bridge variant comparison

Two calibration-time choices in Section 3 are validated by ablation. The linear ridge bridge family beats identity, mean-shift, and residual-MLP variants on held-out residual on both audio and vision. Earlier co-trained bridge variants collapsed toward the identity under the QA loss, so the linear ridge is fit offline and frozen. Encoder-native terse prompt templates beat literary 4-sentence templates by $+0.39$ post-bridge cosine on audio (CLAP) and $+0.28$ on vision (MetaCLIP), so we adopt the terse templates throughout Phase 2.

The deployed bridge of Section 3.3 is a ridge-regularized linear map (Eq. 3). For completeness, Table 12 compares four bridge families on held-out residual $\varepsilon_{m'} = \mathbb{E}\|B_{m'} E^{\text{text}}_{m'}(c) - E^{\text{modality}}_{m'}(o)\|_2$. The linear ridge family achieves the smallest residual on both audio and vision.

Table 12: **Bridge calibration residuals.** Held-out bridge error from text-tower to modality-encoder space; lower is better. The linear bridge gives the lowest residual for audio and vision.

Bridge	Audio	Vision
identity (no bridge)	0.94	8.22
mean-shift	0.86	5.36
residual MLP	0.78	2.12
linear (ridge) ✓	**0.70**	**0.86**

I Additional composition controls

All numbers below are from the reported Qwen3-8B LoRA model at r=512, on the test splits of the given benchmarks.

Training baselines. In Table 4, the "without proxy and bridge" row omits the non-anchor modality entirely, meaning that no token is given, and it is not zeroed or replaced by a learned null. The "without bridge" row keeps the same proxies, but injects them without the learned text-modality mapping the bridge contributes.

Table 13: QA supervision and MSP composition. All numbers are at r=512 on the test splits.

Stage	AVQA	MUSIC-AVQA
Phase 1 only (per-modality projectors)	70.21	45.75
+ QA supervision only, no proxies, no bridge	77.46	55.18
+ full MSP composition	85.58	65.65
QA-supervision contribution	+7.25	+9.43
composition contribution	+8.12	+10.47

Real modalities vs. proxy modalities at inference. We observe that for both benchmarks, highest accuracy is obtained when both modality inputs are real modality samples. Additionally, we see consistent decreases in performance when a single modality is swapped for a proxy, and an even larger decrease when both modality inputs are swapped for generated proxies. All six drops are significant (all $p < 10^{-6}$, paired McNemar test over per-sample predictions from a fixed model). These results are consistent with our view that proxies function as a useful training-time scaffold for learning cross-modal structure, but do not carry the same fidelity of information as their real counterparts.

Token-matched controls. The Gaussian is matched to the real features' per-dimension mean and standard deviation, and shuffled is a genuine encoder output (just one taken from a different test example). Notably, real examples with the wrong features cost the most on both controls, which suggests that a real sample from the same distribution can actively mislead a model when its content does not help answer the given query.

J MUSIC-AVQA-R

As an additional control, we tested our model against MUSIC-AVQA-R [31], which is the debiased re-split of MUSIC-AVQA that contains a tail reweighted toward rarer answers, with the specific goal of defeating answer-prior exploitation. On the full 211,572-question test set we achieved the following (Table 14):

Table 14: MUSIC-AVQA-R results on the full 211,572-question test set.

Split	n	MSP	Majority class
head	138,153	67.26	35.60
tail	73,419	47.76	31.67
overall	211,572	60.49	—

We believe that this result shows additional evidence that MSP learns to use cross-modal structure effectively and can perform well even when compared against models that use direct supervision.

K Sensitivity to proxy quality at test time

We compare the model at its best (all real inputs) with proxies at two levels of quality: (1) scene-grounded proxies are proxies generated from the Phase 1 model by having it observe an anchor modality and imagine what the counterpart modality might be, and (2) a question-derived proxy with no accompanying modality input. A question-derived proxy is generated by asking the model for a possible audio proxy based on the given question's prompt (Table 15).

Table 15: Audio proxy quality at test time with real vision.

Audio proxy quality (vision real)	AVQA (r=512)
real audio	85.58
scene-grounded (imagined from real vision)	81.19
question-derived (observes nothing)	76.15

For example:

> **Question:** "What's driving in the video?"
> **Audio:** `<proxy>` A car engine roars as tires screech and glass shatters `</proxy>`
> **Vision:** `<real>` A blue car drives past a red truck on a sunny highway `</real>`

From the token-matched controls, the table above, and this example (generated by the 8B r=512 model) it is clear why weaker proxies might not function as well as stronger counterparts. In this example there is clearly conflicting information—the audio paints the picture of a car racing and creating some kind of accident, while the vision input specifies a more peaceful highway scene.

This also implies that a multimodal model may not always be necessary for generating high-quality proxies if the generator has a decent enough signal (e.g., detailed captions as input) and sufficient multimodal priors learned from text. This is a future direction worth exploring.

L Paired data at a smaller scale

For this control we took a text-only Qwen3 4B model and finetuned it on AVQA across 5 training paradigms: (1) paired data only, (2) 1 anchor + 1 proxy (general MSP recipe, alternating the anchor modality), (3) interleaved unimodal (1 modality per sample, other slot is left empty), (4) vision only data, and (5) audio only data (Table 16).

Table 16: Paired data at a smaller scale: a text-only Qwen3 4B model finetuned on AVQA.

Method	AVQA accuracy
Paired upper bound (fully paired data)	87.45
MSP (ours)	86.00
Interleaved unimodal (alternating modalities by sample)	81.46
Vision only	80.72
Audio only	75.35

While this does not determine the upper bound of our headline $r{=}512$ model, it does indicate MSP is a strong alternative to fully paired data. The important takeaway is not just the gap between fully paired and MSP (which is relatively small), but the gap between MSP and interleaved unimodal training. This shows that when fully paired data is unavailable, there is an option beyond leaving absent modality slots empty.

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: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. They are also backed by experimental results.

   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: The paper discusses the limitations of the work in Section 5: Limitations and Future Directions.

   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: [N/A].

Justification: The paper does not include theoretical results.

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: The paper fully disclose all the information needed to reproduce the main experimental results including the datasets, model architectures and hyperparameters.

   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: [No].

Justification: Code will be released publicly by the time of the conference at `https://csiplab.github.io/Modality-Self-Play/`. Comprehensive implementation details, including the training schedule and all hyperparameters, are provided in Section 4.1 and Appendix D so that the results can be reproduced independently.

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: The paper specify all the training and test details necessary to understand the results in Section 4.1.

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: [No].

Justification: Due to high computational costs for training, we report results from single runs following standard practice for large-scale multimodal models.

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: Appendix D reports the type of compute workers, the GPU configuration and memory, and the total GPU-hours consumed by the reported training and evaluation runs.

   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: The research conducted in the paper conform, in every respect, with 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: The paper discusses potential societal impacts in Section 5.

    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: [Yes].

    Justification: We discuss potential misuse risks regarding synthetic content generation and propose mitigation strategies in Section 5.

    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: Every dataset, pretrained encoder and language model used in this work is cited, and each asset is used under the license and terms of use of its original release, as stated in Section 4.1.

    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: The paper does not release new assets at this time. Code will be released by the time of the conference.

    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 core method development in this research does not involve LLMs as any important, original, or non-standard components. We used LLM for editing and implementing standard code segments.

    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.