석 사 학 위 논 문
Master's Thesis

멀티모달 학습을 위한 효율적인 훈련 기법

Efficient Training Techniques for Multimodal Learning

2024

이 재 우 (李 在 祐 Lee, Jaewoo)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

멀티모달 학습을 위한 효율적인 훈련 기법

2024

이 재 우

한 국 과 학 기 술 원

김재철AI대학원

멀티모달 학습을 위한 효율적인 훈련 기법

이 재 우

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2024년 06월 17일

심사위원장 황 성 주 (인)

심 사 위 원 이 주 호 (인)

심 사 위 원 윤 세 영 (인)

Efficient Training Techniques for Multimodal Learning

Jaewoo Lee

Advisor: Sung Ju Hwang

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in AI

Daejeon, Korea
June 17, 2024

Approved by

Sung Ju Hwang
Professor in the Kim Jaechul Graduate School of AI

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MAI 이재우. 멀티모달 학습을 위한 효율적인 훈련 기법. 김재철AI대학원 . 2024년. 60+v 쪽. 지도교수: 황성주. (영문 논문)
Jaewoo Lee. Efficient Training Techniques for Multimodal Learning. Kim Jaechul Graduate School of AI . 2024. 60+v pages. Advisor: Sung Ju Hwang. (Text in English)

초 록

끊임없이 변화하는 세상에서 모델이 새로운 지식을 습득하는 것은 모델의 성능을 유지하는데 중요한 요소이다. 본 학위논문에서는 멀티모달 모델을 최신 상태로 유지하는 효율적인 방법을 다루었다.

첫 번째로, 새로운 의미를 가진 오디오-비디오 데이터의 분포가 시간이 지남에 따라 지속적으로 변하는 상황에서 모델을 사전 학습시키는 연속 학습 방법을 다루었다. 구체적으로, 우리는 경량 인코더를 사용하여 (1) 크로스-어텐션을 통해 오디오-비디오 패치의 중요성을 측정하고 (2) 현재 오디오와 과거 오디오 또는 현재 비디오와 과거 오디오 간의 멀티모달 상관관계를 평가한다. 이 방법은 현재 데이터에서 과거 오디오-비디오 의미와 높은 상관관계를 보이면서도 과거 오디오-비디오 정보와 낮은 상관관계를 보이는 오디오-비디오 패치를 식별한다. 따라서 이 접근 방식은 타깃 모델을 지속적으로 사전 학습시키면서 과거 오디오-비디오 지식을 잊는 것을 최소화하고 GPU 메모리 사용량을 크게 줄인다.

두 번째로, 새로운 대형 비전-언어 모델 (LVLM)의 개발 비용을 줄이기 위해 시각적 지시 튜닝 (VIT)에 대한 코어셋 선택 기법을 다루었다. 작은 비전-언어 모델의 내부 활성화 정보를 활용하여 VIT 데이터를 LVLM 일반화에 필요한 시각-언어적 개념-기술 구성으로 클러스터링한다. 그 후, 앞서 구한 다양한 클러스터에서 클러스터의 밀도와 전이 가능성을 고려하여 데이터를 선별한다. 이 전략은 선별된 데이터의 훈련 효율성을 높이고 코어셋 내에서 높은 개념-기술 구성의 다양성을 보장함을 통해 기존의 전체 VIT 데이터셋의 불필요한 중복을 줄여 학습 효율성을 높인다.

핵 심 낱 말 효율적 훈련, 멀티모달 학습, 계속적 학습, 데이터 프루닝

Abstract

Maintaining model performance requires updating them with new knowledge from our ever-evolving world. My thesis focuses on efficient techniques to keep multimodal models up-to-date.

Firstly, we propose a continual learning method that pre-trains models with audio-video data whose distribution continuously changes over time with new semantics. Specifically, we employ a lightweight encoder that (1) estimates the importance of audio-video patches using cross-attention and (2) assesses the multimodal correlation between the current audio and the past video or the current video and the past audio. This method identifies semantically intertwined audio-video patches from current data while showing low correlation with the past audio-video semantics. Consequently, this approach allows us to continually pre-train target models while minimizing the forgetting of past audio-video knowledge and significantly reducing GPU memory usage.

Secondly, we introduce a coreset selection technique for visual instruction tuning (VIT) to reduce the development cost of new Large Vision-Language Models (LVLMs). By leveraging the inner activations of a small Vision-Language Model, we cluster VIT data into fine-grained visual-linguistic concept-skill compositions, which LVLM needs for its generalization. We then sample data from these diverse clusters by considering their density and transferability. This strategy enhances training efficacy and ensures high concept-skill diversity within the coreset, thereby reducing the redundancy of the original VIT dataset to enhance efficiency in LVLM finetuning.

Keywords Efficient Training, Multimodal Learning, Continual Learning, Data Pruning

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Multimodal learning is essential for numerous real-world applications due to the widespread presence of data types like text-image, text-video, and audio-video pairs. However, current multimodal learning methods struggle in real-world scenarios where training data continuously changes over time with new multimodal content. This constant change leads to trained models becoming outdated quickly. A common solution to keep models up-to-date is to periodic retraining from scratch using the latest training data. However, this approach is highly demanding in terms of computational power and memory.

In Chapter 2, we propose a continual audio-video pre-training framework that can continuously learn audio-video semantics from the ever-evolving audio and video data distributions. We first introduce two critical challenges in this scenario: *sparse spatio-temporal correlation* between audio-video pairs and *multimodal correlation overwriting* that leads to forgetting previously learned audio-video relations. To address these challenges, we propose two novel ideas: *(1) Localized Patch Importance Scoring*: we introduce a multimodal encoder to determine the importance score for each patch, emphasizing semantically intertwined audio-video patches. *(2) Replay-guided Correlation Assessment*: to reduce the corruption of previously learned audiovisual knowledge due to drift, we propose assessing the correlation of current patches with past steps to identify patches exhibiting high correlations with previous steps. We experimentally demonstrate that our method enables continuous pre-training of models with new audio-video semantics, resulting in better performance across various audiovisual downstream tasks. Additionally, our method enhances efficiency during pre-training in terms of GPU memory consumption and training time.

In Chapter 3, we present an efficient coreset selection technique designed for visual instruction tuning (VIT) aimed at reducing the development cost of Large Vision-Language Models (LVLMs). Conventional coreset selection techniques (1) do not adequately cover the diversity of vision-language (VL) tasks in VIT datasets and (2) are computationally expensive or require advanced models during the coreset selection procedure. In contrast, our method leverages a small model as a reference model to strategically select diverse and transferable VIT data for finetuning a target LVLM. We use the inner activations from the small model to cluster the training data into fine-grained VL concept-skill compositions, crucial for the target LVLM's generalization. By sampling from these clusters based on their density and transferability, we enhance training efficacy while ensuring a high diversity of concept-skill compositions within the coreset. Our extensive experiments demonstrate that our approach significantly enhances performance and efficiency. It achieves competitive performance compared to LVLMs finetuned on the full VIT dataset, while substantially reducing the amount of data required. This underscores our method's effectiveness and scalability in optimizing the finetuning process for LVLMs.

This Chapter is based on the work that is published at ICML 2024 [8].

Chapter 2. Continual Audio-Video Pre-training with SpatioTemporal Localized Alignment

Continuously learning a variety of audio-video semantics over time is crucial for audio-related reasoning tasks in our ever-evolving world. However, this is a nontrivial problem and poses two critical challenges: *sparse spatio-temporal correlation* between audio-video pairs and *multimodal correlation overwriting* that forgets audio-video relations. To tackle this problem, we propose a new continual audio-video pre-training method with two novel ideas: *(1) Localized Patch Importance Scoring*: we introduce a multimodal encoder to determine the importance score for each patch, emphasizing semantically intertwined audio-video patches. *(2) Replay-guided Correlation Assessment*: to reduce the corruption of previously learned audiovisual knowledge due to drift, we propose to assess the correlation of the current patches on the past steps to identify the patches exhibiting high correlations with the past steps. Based on the results from the two ideas, we perform probabilistic patch selection for effective continual audio-video pre-training. Experimental validation on multiple benchmarks shows that our method achieves a 3.69%p of relative performance gain in zero-shot retrieval tasks compared to strong continual learning baselines, while reducing memory consumption by ∼45%.

2.1 Introduction

Multimodal learning is an important problem for various real-world applications, as many real-world data types are multimodal, such as *text-image* [9, 10], *text-video* [11, 12], and *audio-video* [13, 14] pairs. While most vision-language learning [15, 16, 17] assumes the availability of curated multimodal data with human-annotated descriptions, audiovisual domain [18, 19] holds a unique and practical advantage, as most videos inherently come with accompanying audios without human annotations. Thanks to this property, audiovisual multimodal learning models can leverage web-scale raw videos (e.g., YouTube, TikTok, etc.) for training with minimal human efforts in data preprocessing, and thus have achieved impressive success in audio-video compositional reasoning [20, 21, 22].

However, most existing approaches [20, 21, 19] struggle when deployed to real-world scenarios, where **the distribution of training data continuously changes over time with new audio-video semantics.** For example, the audiovisual model pre-trained before electric vehicles became popular, would not be able to associate *cars* with their unique acoustic cues (e.g., motor sound) (See Figure 2.1). One straightforward solution to this problem is to periodically train the model from scratch using audio-video data collected from the past to the present, but this approach comes with prohibitive computation and memory costs.

While continual learning is a viable solution for tackling such scenarios, dealing with dynamically evolving audio-video semantics is a nontrivial problem due to two critical challenges. First, the spatio-temporal correlation between the audio-video data is highly sparse. As represented in Figure 2.3 (b), only a few objects/regions in a video (i.e., sound sources) are strongly correlated with audio. Secondly, audio-video pre-training models encounter the issue of forgetting not only the representations of each modality but also the correlation between them. As *orange circles* in Figure 2.3 (c) illustrate, the model

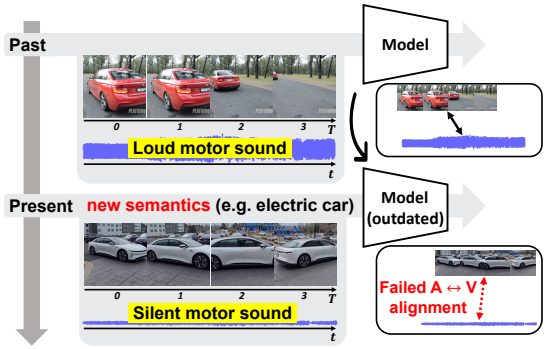


Figure 2.1: **Outdated pre-trained audio-video models.** The outdated models struggle with understanding emerging new audio-video semantics.

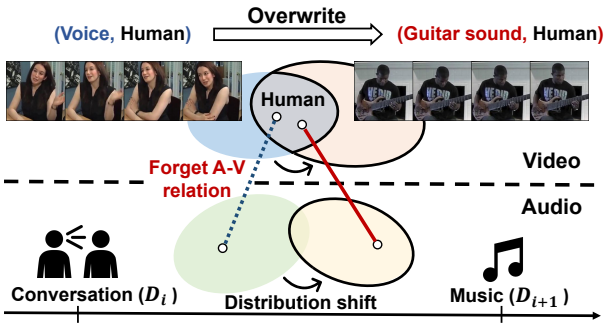


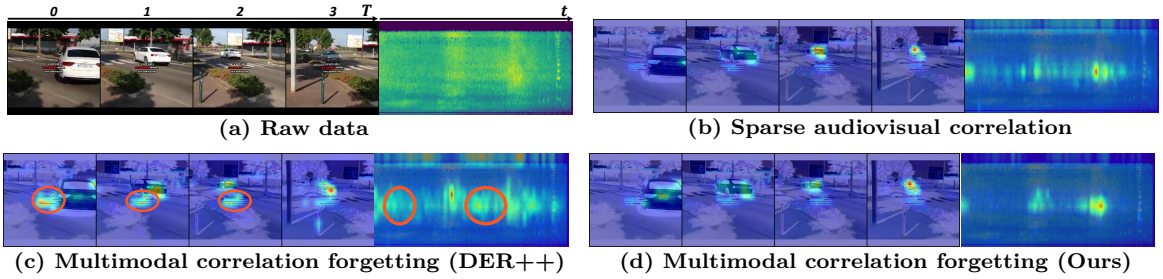
Figure 2.2: **Challenge of multimodal correlation overwriting.** During continual pre-training, the model can encounter new semantics sharing key visual objects, humans, making the model overwrite the previously learned human voice audio information (blue) to a new one (i.e., guitar) (red), resulting in forgetting.

which initially learned the accurate audio-video correlation in a car's engine video, forgets this correlation after learning on a series of audio-video tasks. It instead highlights inaccurate regions in the audio-video data, as if there were highly fine-grained multimodal alignment.

To overcome these challenges in learning multiple audio-video tasks sequentially, we propose ***S**patio-**TE**mporal **L**oca**L**ized **A**lignment (**STELLA**)*, a novel approach that exploits past and current information via audio-video attention maps. Specifically, our goal is to continually pre-train the model by selecting audio and video patches that have a high correlation for its modality pair and also preserve previously learned audio-video correlation. Thereby we propose a probabilistic patch selection framework that enables the model to learn better audio-video correlations and preserve past audio-video semantics, based on two key components: first, we use the averaged cross-attention maps obtained by a lightweight multimodal encoder to compute an importance score, estimating how each audio (or video) patch is important for its modality pair. Further, to preserve the past correlation during continual audio-video pre-training, we leverage new cross-attention maps activated by the key and query embeddings between the current and past steps, respectively. This yields a correlation score that identifies the patches that exhibit a higher correlation with the current step than the past steps. We extensively validate our method on continual audio-video pre-training scenarios, using diverse benchmark datasets evaluated on various audiovisual downstream tasks. Our method outperforms strong baseline on various tasks with enhanced efficiency by reducing the GPU memory by ∼45% during continual pre-training. We further provide extensive in-depth analysis with visualizations.

Our paper makes the following key contributions:

- We are the first to address continual audio-video pre-training, which poses new challenges: *sparse spatio-temporal correlation* between audio-video pairs and *multimodal correlation overwriting* that forgets their relations.
- We propose a novel method that leverages cross-attention maps to capture sparse audio-video relationships and mitigate forgetting of previously learned relationships.
- We demonstrate the efficacy of our method on several audiovisual downstream tasks including retrieval, sound source localization and event localization. In particular, ours achieves 3.69%p of performance gain in the retrieval task and reduces the GPU memory consumption by ∼45% during training, compared to the strongest baseline.



(a) Raw data **(b) Sparse audiovisual correlation**

(c) Multimodal correlation forgetting (DER++) **(d) Multimodal correlation forgetting (Ours)**

Figure 2.3: **Challenges in continual audio-video learning. (a):** A raw data pair describing a car and its engine sound. **(b):** Sparse correlations in cross-attention maps. **(c):** After training on a series of tasks after (b), *DER++* focuses on entirely different areas (orange circle), presenting correlation forgetting. **(d):** Our *STELLA* maintains consistent attention.

2.2 Related Work

Audiovisual understanding Self-supervised learning on audiovisual data aims to learn transferable representations that can be applied to a variety of audio-image/video downstream tasks, including action recognition/event classification [23, 24], sounding object localization [25, 26], and multimodal retrieval [21, 19]. Inspired by the success of Masked AutoEncoders (MAE) in visual pre-training [27], recent audiovisual representation learning adopts masked modeling for comprehending audiovisual semantics [20, 19]. TVLT [20] adopts the MAE structure and audiovisual matching to predict whether audio and visual data originated from the same video. CAV [19] combines the MAE with audiovisual contrastive learning, which pulls matching audiovisual pairs closer and pushes non-matching pairs apart. Their methods assume a fixed input data distribution that does not shift throughout training. However, in the real world, a machine/agent will continuously encounter new (i.e., changing distribution) audio-video tasks/semantics. If not well managed, the methods will suffer severe performance degradation if they encounter the aforementioned shift in continual learning, a challenging and realistic scenario for multimodal learning.

Multimodal continual learning Continual learning [28, 29, 30] refers to a learning paradigm in which a model sequentially learns an unlimited number of tasks/domains. It aims to continuously adapt to new tasks while preserving previously learned knowledge/skills, which is crucial for real-world AI deployment. A number of works have addressed supervised learning for vision tasks [31, 32, 33], and very recently, a few approaches have explored continual learning with self-supervised learning [34, 35, 36, 37], and multimodal learning [38, 39, 40]. AV-CIL [39] and CIGN [40] tackle the problem of supervised continual learning for audio-video tasks. However, they require dense human annotations, such as text or audiovisual labels, and task boundary information to know when new tasks are introduced during continual learning. On the other hand, our *STELLA* focuses on continual pre-training of audio-video models without any human-effort labels or task boundary information. Moreover, our work extends to investigating the impact of past data on the current audio and video attention map activation, while the AV-CIL focuses on maintaining the past visual attention map.

2.3 Continual Audio-Video Pre-training

2.3.1 Problem Statement

In this work, we tackle the problem of continual audio-video pre-training, under the assumption that the data distribution continuously changes during pre-training, and the model does not have direct access

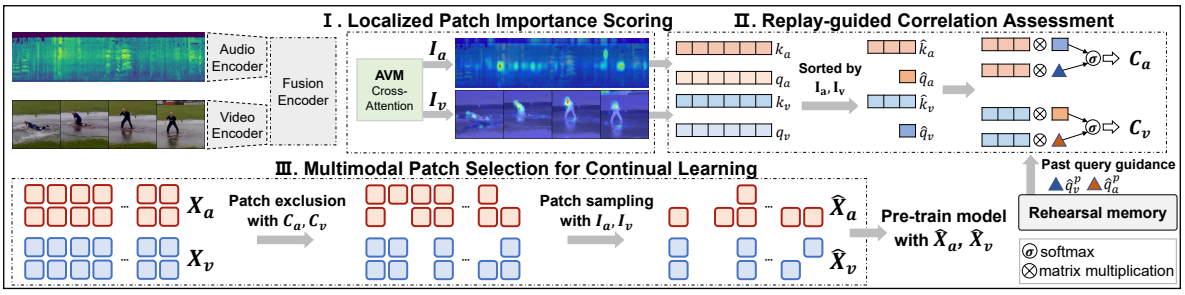


Figure 2.4: **Overview of our approach.** Our method harnesses cross-modal attention maps from the AVM module to compute importance scores in order to identify highly correlated patches (**Localized Patch Importance Scoring**). Comparing the attention maps created by the current queries with those generated by past queries, we compute correlation scores of the current patches with the past data (**Replay-guided Correlation Assessment**). Finally, we perform a probabilistic patch selection, combining the importance scores and correlation scores to select patches for continual audio-video pre-training (**Multimodal Patch Selection for Continual Learning**).

to previously seen data and stores only a small subset in the rehearsal memory [41, 42]. Furthermore, we assume a task-free scenario [43] where the model performs the pre-training and inference without the explicit knowledge of task boundaries, which is challenging yet realistic as the model does not need any human guidance on the change of data distributions. Following the setup in continual learning literature [34, 44], we formulate pre-training of the audio-video learning model over a sequence of $\mathcal{T}$ disjoint audio-video datasets $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{\mathcal{T}}$. For the i-th task, the model iteratively samples B audio-video pairs $(X_a^i, X_v^i) \sim \mathcal{D}_i$[1]. Here, $X_a \in \mathbb{R}^{B\times M\times p\times p}$ represents the audio patches, patchfied from the audio spectrogram with time (t) and frequency (f) dimensions, where $M = |t/p| \cdot |f/p|$ and p is the patch size. Similarly, $X_v \in \mathbb{R}^{B\times N\times p\times p}$ represents the video patches, obtained from the video clip with channel, frames (T), height (h), and width (w) dimensions, where $N = |T| \cdot |h/p| \cdot |w/p|$.

Following [19], the model $f(\cdot;\boldsymbol{\theta})$ comprises audio-video encoders, a multimodal fusion encoder, and a decoder. For pre-training, we adopt two loss terms: *reconstruction loss* (ℓ^r) for masked patches to understand low-level audio-video features, and *masked contrastive loss* (ℓ^c) for pooled audio-video features to learn semantic relationships between the two. During each training iteration for task i, the model updates weights by minimizing the objective $\mathcal{L} = \ell^r(f_{\boldsymbol{\theta}}(\mathcal{D}_i)) + \lambda\ell^c(f_{\boldsymbol{\theta}}(\mathcal{D}_i))$, with a balancing term λ. The detailed mathematical expressions of the loss functions are explicated in Equation 2.10. Then, we evaluate the learned representations through various audiovisual downstream tasks at the end of the task.

2.3.2 Challenges in Continual Audio-Video Pre-training

In this section, we delve into two key challenges in continual audio-video pre-training: 1) *sparse spatio-temporal correlation* 2) *multimodal correlation overwriting*. In Figure 2.3 (b), we visualize cross-attention heat maps and observe *sparse spatio-temporal correlation* between the audio-video pair. Capturing highly correlated audio-video patches is crucial for understanding their semantics, allowing the model to focus on informative regions and learn complex multimodal relationships. It becomes more critical in continual audio-video pre-training methods in view of *rehearsal memory*. They contain a small-sized rehearsal memory designed to store key information for past tasks during continual pre-training. As rehearsal memory is limited in capacity, it's important to store meaningful data/feature audio-video pairs associated with their semantics.

[1] We omit the task index for brevity, unless otherwise stated.

We also observe that the model forgets previously learned audio-video correlations after learning a sequence of tasks (Figure 2.3 (c)). In continual audio-video pre-training, the biased data distribution poses a risk of overwriting previous multimodal correlations, driven by the close correlation between current video and past audio data, and vice versa. For instance, transitioning from a past task involving human-conversational data to a current task featuring human-playing-musical-instrument data (Figure 2.2) weakens the audio-video correlations of human visuals and voices from the past task. Instead, the model potentially associates human visuals with musical sounds prevalent in the biased current data distribution, leading to the forgetting of the past human-voice relationships. This challenge, termed *multimodal correlation overwriting*, underscores the critical need to identify data regions with high correlation to past steps.

2.4 Continual Audio-Video Pre-training with Spatio-Temporal Localized Alignment

To overcome critical challenges in earlier sections, we introduce a novel continual audio-video pre-training approach, dubbed ***S**patio-**TE**mporal **L**oca**L**ized **A**lignment (**STELLA**)*, illustrated in Figure 2.4. We first propose a lightweight trainable module that determines importance scores, guiding the model to focus on spatio-temporally aligned audio-visual regions (Section 2.4.1). Next, we introduce a unique process of assessing multimodal correlations between current and previous steps to compute correlation scores, identifying patches having higher correlations to the past steps (Section 2.4.2). Finally, we describe the probabilistic patch selection framework, which uses the importance and correlation scores to select audio and video patches for continual pre-training (Section 2.4.3). Please see Algorithm 2 for a detailed training process.

2.4.1 Localized Patch Importance Scoring

Inspired by the observation that audio-video data pairs are only correlated with a sparse spatio-temporal region, we aim to capture accurate local semantics between audio and visual cues by computing importance scores for each patch to identify a few strongly associated audio-video patches. We achieve this by introducing an Audio-Video Matching (AVM) module that uses cross-attention to capture core audio-video patches. Given (X_a, X_v), we first map audio/video patches using the modality encoders and fusion encoder to output tokens $(\boldsymbol{o}_a, \boldsymbol{o}_v)$. Then, we fed the tokens to the AVM module to map them to queries and keys $(\boldsymbol{q}, \boldsymbol{k})$ to compute cross-attention maps as follows:

$$\begin{aligned} \boldsymbol{q}_a = \boldsymbol{o}_a \mathcal{W}_a^Q,\ \boldsymbol{k}_a = \boldsymbol{o}_a \mathcal{W}_a^K,\ \boldsymbol{q}_v = \boldsymbol{o}_v \mathcal{W}_v^Q,\ \boldsymbol{k}_v = \boldsymbol{o}_v \mathcal{W}_v^K, \\ \boldsymbol{A}_a = \mu(\boldsymbol{q}_v, \boldsymbol{k}_a) = \boldsymbol{q}_v \boldsymbol{k}_a^\top / \beta * \sqrt{d}, \\ \boldsymbol{A}_v = \mu(\boldsymbol{q}_a, \boldsymbol{k}_v) = \boldsymbol{q}_a \boldsymbol{k}_v^\top / \beta * \sqrt{d}, \end{aligned} \tag{2.1}$$

where the projections $\mathcal{W}_a^Q$, $\mathcal{W}_a^K$, $\mathcal{W}_v^Q$, $\mathcal{W}_v^K \in \mathbb{R}^{D \times H \times d}$ are trainable parameter matrices in the AVM module, H is the number of heads, $D = H * d$ is the dimension size, β denotes a temperature coefficient, $(\boldsymbol{q}_a, \boldsymbol{k}_a) \in \mathbb{R}^{B \times H \times M \times d}$, $(\boldsymbol{q}_v, \boldsymbol{k}_v) \in \mathbb{R}^{B \times H \times N \times d}$ are audio and video keys and queries, $\boldsymbol{A}_a \in \mathbb{R}^{B \times H \times N \times M}$, $\boldsymbol{A}_v \in \mathbb{R}^{B \times H \times M \times N}$ are computed cross-attention maps. Please see Figure 2.11 for the detailed architecture of the AVM module.

Then, we compute the importance scores $\boldsymbol{I}_a \in \mathbb{R}^{B\times M}$, and $\boldsymbol{I}_v \in \mathbb{R}^{B\times N}$ by applying Softmax normalization on the last dimension:

$$\begin{aligned} \boldsymbol{I}_a &= \texttt{MeanPool}\left(\texttt{Softmax}\left(\boldsymbol{A}_a\right)\right), \\ \boldsymbol{I}_v &= \texttt{MeanPool}\left(\texttt{Softmax}\left(\boldsymbol{A}_v\right)\right). \end{aligned} \tag{2.2}$$

The importance score represents the average correlation between an audio (or a video) patch and the paired modality patches. That is, the higher value in $\boldsymbol{I}$ indicates the higher importance of the corresponding patch in view of the opposite modality (A↔V), thus helping the model to select locally aligned audio-video patches in Section 2.4.3.

2.4.2 Replay-guided Correlation Assessment

To tackle the challenge of *multimodal correlation overwriting*, the model requires a careful balance between retaining previous knowledge and adapting new one. Thus, we propose to compare cross-attention maps activated by current and past queries to assess relative multimodal correlation and exclude patches exhibiting higher correlation to the past steps. Our ultimate goal is to select κ_a audio and κ_v video patches where $\kappa_a = M\cdot\rho_a$ and $\kappa_v = N\cdot\rho_v$, with ρ_a and ρ_v denoting sampling ratios for audio and video. To this end, we obtain locally aligned queries $\widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{q}}_v \in \mathbb{R}^{B\times H\times d}$ and keys $\widehat{\boldsymbol{k}}_a \in \mathbb{R}^{B\times H\times \kappa_a\times d}, \widehat{\boldsymbol{k}}_v \in \mathbb{R}^{B\times H\times \kappa_v\times d}$ using the indices sorted in ascending order based on the importance scores $\boldsymbol{S}_a = \texttt{argsort}(\boldsymbol{I}_a)$, $\boldsymbol{S}_v = \texttt{argsort}(\boldsymbol{I}_v)$:

$$\begin{aligned} \widehat{\boldsymbol{q}}_n[i,:,j] = \boldsymbol{q}_n[i,:,\boldsymbol{S}_n[i,j]],\ \boldsymbol{I}^s_n[i,j] = \boldsymbol{I}_n[i,\boldsymbol{S}_n[i,j]], \\ \widehat{\boldsymbol{q}}_n \leftarrow \texttt{MeanPool}\left(\widehat{\boldsymbol{q}}_n, \texttt{weight} = \boldsymbol{I}^s_n\right), \\ \widehat{\boldsymbol{k}}_n[i,:,j] = \boldsymbol{k}_n[i,:,\boldsymbol{S}_n[i,j]],\ i = 1,\dots,B,\ j = 1,\dots,\kappa_n, \end{aligned} \tag{2.3}$$

where $n \in (a, v)$ and $\texttt{MeanPool}\left(\cdot, \texttt{weight}\right)$ indicates weighted mean operation. We utilize the queries and keys to compute cross-attention maps $\widehat{\boldsymbol{A}}_a = \mu_(\widehat{\boldsymbol{q}}_v, \widehat{\boldsymbol{k}}_a) \in \mathbb{R}^{B\times H\times \kappa_a}$, $\widehat{\boldsymbol{A}}_v = \mu(\widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{k}}_v) \in \mathbb{R}^{B\times H\times \kappa_v}$. Similarly, we compute cross-attention maps $\widehat{\boldsymbol{A}}^p_a = \mu(\widehat{\boldsymbol{q}}^p_v, \widehat{\boldsymbol{k}}_a)$,$\widehat{\boldsymbol{A}}^p_v = \mu(\widehat{\boldsymbol{q}}^p_a, \widehat{\boldsymbol{k}}_v)$ by using the past queries $\widehat{\boldsymbol{q}}^p_a$, $\widehat{\boldsymbol{q}}^p_v$, which were computed during the past steps and stored in the rehearsal memory. Each $\widehat{\boldsymbol{A}}$ shows how the given queries are correlated to the current patches. To assess the relative correlation between the past and current steps on the current patches, we stack the audio $(\widehat{\boldsymbol{A}}_a, \widehat{\boldsymbol{A}}^p_a)$ and video attention maps $(\widehat{\boldsymbol{A}}_v, \widehat{\boldsymbol{A}}^p_v)$, resulting in an extended last dimension, respectively. Subsequently, we apply Softmax normalization on the extended last dimension, resulting in correlation scores $\boldsymbol{C}_a$ and $\boldsymbol{C}_v$ as follows:

$$\begin{aligned} \boldsymbol{C}_a &= \texttt{MeanPool}\left(\texttt{Softmax}\left([\widehat{\boldsymbol{A}}_a, \widehat{\boldsymbol{A}}^p_a]\right)\right), \\ \boldsymbol{C}_v &= \texttt{MeanPool}\left(\texttt{Softmax}\left([\widehat{\boldsymbol{A}}_v, \widehat{\boldsymbol{A}}^p_v]\right)\right). \end{aligned} \tag{2.4}$$

Each value in the correlation score moves closer to *one* when the corresponding patch exhibits a higher multimodal correlation with the opposite modality data from the past steps compared to the correlation with its modality pair. Hence, patches with high $\boldsymbol{C}$ values should more likely be excluded to preserve previously learned multimodal correlations.

2.4.3 Multimodal Patch Selection for Continual Learning

Leveraging the importance score $\boldsymbol{I}_v$ and correlation score $\boldsymbol{C}_v$, we enhance multimodal alignment and stability of the continual pre-training by sorting video patch indices. Initially, a Bernoulli distribution on $\boldsymbol{C}_v$ produces $\boldsymbol{F}_v$. True values in $\boldsymbol{F}_v$ indicate that the corresponding patches are chosen to be excluded. Hence, we zero out elements in $\boldsymbol{I}_v$ aligned with the True values in $\boldsymbol{F}_v$ to create $\tilde{\boldsymbol{I}}_v$. Subsequently, applying a multinomial probability distribution to $\tilde{\boldsymbol{I}}_v$ yields the informative video patch indices $\tilde{\boldsymbol{S}}_v \in \mathbb{R}^{B\times N}$:

$$\tilde{\boldsymbol{I}}_v[i,j] = \begin{cases} 0 & \text{if } \boldsymbol{F}_v[i,j] \quad i=1,\ldots,B \\ \boldsymbol{I}_v[i,j] & \text{otherwise} \quad j=1,\ldots,N, \end{cases} \tag{2.5}$$
$$\tilde{\boldsymbol{S}}_v = \texttt{Multinomial}\left(\tilde{\boldsymbol{I}}_v,\right).$$

Similarly, we utilize the importance score $\boldsymbol{I}_a$ and correlation score $\boldsymbol{C}_a$ to generate the informative audio patch indices. To preserve the local correlation among audio patches by temporal continuity, we segment $\boldsymbol{I}_a$ into time chunks. To this end, we reshape the importance score $\boldsymbol{I}_a$ into a time-frequency dimension, average along the frequency dimension, and split the time dimension with time chunk size $\boldsymbol{L}_c$. This operation yields $\boldsymbol{I}_a^c \in \mathbb{R}^{B\times|t/p|/|\boldsymbol{L}_c|}$, which indicates the importance of audio time chunks. For $\boldsymbol{C}_a$, we apply Bernoulli probability distribution to generate $\boldsymbol{F}_a$.

We select informative time chunks with high $\boldsymbol{I}_a^c$ values while excluding the indices aligned with True values in $\boldsymbol{F}_a$ to generate the informative audio patch indices $\tilde{\boldsymbol{S}}_a \in \mathbb{R}^{B\times M}$. The detailed steps of audio patch selection are in Algorithm 1.

Finally, based on $\tilde{\boldsymbol{S}}_a, \tilde{\boldsymbol{S}}_v$, we select κ_a, κ_v of audio, video patches to form new input $(\widehat{X}_a, \widehat{X}_v)$. Substituting (X_a, X_v) into $(\widehat{X}_a, \widehat{X}_v)$ enables the model to effectively learn new audio-video relationships while preserving previously learned ones with enhanced efficiency. The final patch selection is performed as follows:

$$\widehat{X}_n[i,j] = X_n[i, \tilde{\boldsymbol{S}}_n[i,j]],\ i=1,\ldots,B,\ j=1,\ldots,\kappa_n, \tag{2.6}$$

where $n \in (a,v)$. With the selected patches, we perform continual pre-training based on the *DER++* framework with the penalty loss (ℓ^p), which encourages the model to maintain the features of the rehearsal memory to mitigate their drifts. Hence, our final pre-training objective is $\mathcal{L} = \ell^r + \lambda\ell^c + \alpha\ell^p$, where α is a hyperparameter for the penalty loss.

Efficient rehearsal memory usage is crucial especially in continual audio-video learning scenarios due to the large video sizes. The effective storage of past data can notably augment the diversity of data within the memory. To address this, we propose *STELLA+*, an extension of *STELLA*, where memory stores the selected patches instead of raw data (Algorithm 3). The introduction of *STELLA+* represents a distinct and complementary direction to *STELLA*, demonstrating the efficacy of efficient memory utilization.

2.5 Experiments

In this section, we experimentally validate the effectiveness of our method in task-free continual audio-video pre-training. We start by outlining our experimental setup in Section 2.5.1, covering datasets, evaluation methods, evaluation metrics, and baseline methods employed for our experiments. Subsequently, we present the experimental results and conduct a comprehensive analysis in Section 2.5.2.

Table 2.1: **Zero-shot retrieval results.** Results of audiovisual zero-shot retrieval task on *Continual-VS* and *Continual-AS*. R@K means top-K recall. The results are the means of 3 independent runs. The best and the second best results are highlighted in **bold** and underline, respectively.

	Method	Continual-VS R@1		R@5		R@10		Avg		Continual-AS R@1		R@5		R@10		Avg	
		$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
Audio-to-Video	Finetune	0.98	4.16	3.75	11.98	6.17	15.35	3.63	10.50	1.48	2.90	3.84	11.34	5.41	17.83	3.58	10.69
	ER	4.09	3.66	11.66	9.17	17.78	10.20	11.18	7.68	4.94	2.97	12.33	7.46	17.60	11.17	11.62	7.20
	MIR	4.59	3.14	12.26	8.34	17.51	11.17	11.45	7.55	5.21	2.93	13.16	7.10	18.04	9.14	12.14	6.39
	DER++	4.03	3.62	13.74	6.31	19.79	7.11	12.52	5.68	4.51	3.75	12.15	8.42	16.85	11.86	11.17	8.01
	GMED	4.17	2.73	12.01	6.84	18.95	6.33	11.71	5.30	4.71	2.27	12.83	7.45	18.44	9.18	11.99	6.30
	CLS-ER	4.61	3.20	14.07	6.77	19.54	8.92	12.74	6.30	4.17	4.50	11.28	11.06	16.85	12.55	10.77	9.37
	LUMP	3.56	2.79	11.68	7.65	17.40	8.52	10.88	6.32	3.73	3.03	13.74	**5.29**	19.50	8.17	12.32	**5.50**
	ESMER	4.51	3.68	14.98	6.22	21.25	7.50	13.58	5.80	5.18	4.92	14.14	9.19	18.69	12.84	12.67	8.98
	STELLA (Ours)	5.34	**2.04**	15.04	5.20	22.10	**5.90**	14.16	**4.38**	5.22	**2.26**	13.09	7.95	18.75	10.65	12.35	6.95
	STELLA+ (Ours)	**5.39**	2.71	**16.76**	**5.15**	**24.18**	5.99	**15.44**	4.62	**5.36**	4.24	**16.76**	5.54	**23.65**	**7.44**	**15.26**	5.74
	Multitask	6.45	–	20.19	–	29.01	–	18.55	–	8.28	–	24.14	–	33.74	–	22.05	–
Video-to-Audio	Finetune	1.22	4.47	4.17	11.23	6.95	14.67	4.11	10.12	1.50	**3.23**	4.08	10.04	6.33	14.43	3.97	9.23
	ER	3.28	3.94	11.30	8.86	16.40	11.37	10.33	8.06	3.70	4.36	10.76	10.34	15.68	15.06	10.05	9.92
	MIR	3.54	3.47	11.82	9.11	16.69	12.90	10.68	8.49	4.26	4.59	11.29	9.87	15.97	13.73	10.51	9.40
	DER++	3.49	3.86	13.22	7.09	19.03	9.04	11.91	6.66	4.23	4.50	11.66	10.10	16.24	13.97	10.71	9.52
	GMED	3.71	2.61	11.87	6.46	17.20	9.57	10.93	6.21	3.99	4.42	10.65	10.39	15.41	14.78	10.02	9.86
	CLS-ER	4.09	3.11	13.30	6.96	19.43	9.68	12.27	6.58	4.25	4.58	9.78	11.65	13.45	17.65	9.16	11.29
	LUMP	3.24	3.30	11.02	7.55	16.91	9.13	10.39	6.66	3.13	3.91	10.60	8.63	16.02	12.26	9.92	8.27
	ESMER	4.65	2.74	14.54	6.27	20.80	8.36	13.33	5.79	4.39	4.92	11.55	12.16	16.41	16.41	10.78	11.16
	STELLA (Ours)	5.30	2.40	15.43	4.84	21.47	6.70	14.07	4.65	4.49	3.39	12.08	9.00	17.31	12.75	11.29	8.38
	STELLA+ (Ours)	**5.86**	**1.56**	**17.21**	**4.09**	**23.53**	**6.02**	**15.53**	**3.89**	**5.48**	4.06	**15.65**	**7.13**	**22.29**	**8.92**	**14.47**	**6.70**
	Multitask	6.85	–	21.93	–	30.63	–	19.80	–	8.05	–	25.81	–	35.60	–	23.15	–

2.5.1 Experimental Setup

Evaluation Protocol We validate our method on continual audio-video pre-training over VGGSound [45] and AudioSet [46] datasets, consisting of 10s videos. We split each dataset into multiple tasks based on its high-level category information. We name them as *Continual-VS* and *Continual-AS*, respectively. For evaluation, we conduct various audiovisual downstream tasks: retrieval, sound source localization, and event localization. Further details, including data split, data statistics, and downstream tasks, are provided in Section 2.9.

Baselines To quantitatively assess our method, we compare its performance with several task-free continual learning methods: ER [41], MIR [47], DER++ [42], GMED [48], CLS-ER [49], LUMP [34], and ESMER [44]. The details of the baseline methods are explicated in Section 2.8. All methods employ reservoir sampling [50] to sample past instances from the rehearsal memory for $2K$ (*Continual-VS*) and $5K$ (*Continual-AS*) instances during continual pre-training, except for *STELLA+*, which adjusts instance count based on sampling ratios (ρ_a, ρ_v) to match the memory size of other methods. We additionally report the result of *Finetune*, the model continually pre-trained without additional methods, and *Multitask*, the model pre-trained with the entire datasets. They serve as lower and upper bounds, respectively, in assessing learned representation.

Evaluation Metrics After each end of pre-training on $\mathcal{D}_t$, we estimate task-specific performances $\{acc_{t,i}\}_{i=1}^{t}$, where $acc_{t,i}$ denotes the performance of the downstream task associated with $\mathcal{D}_i$ when evaluated with $f_{\theta,t}$, the model pre-trained up to the t-th task. Here, no task boundary information is employed in performance estimation. For the evaluation, we adopt two conventional metrics in continual learning: **(1) Average accuracy**($\mathcal{A}$) is the mean accuracy across all tasks after the completion of pre-training on $\mathcal{D}_\mathcal{T}$, and it is formulated as $\mathcal{A} = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} acc_{\mathcal{T},i}$. **(2) Average Forgetting**($\mathcal{F}$) measures the average amount of catastrophic forgetting for each task, quantified as the difference between its maximum accuracy and accuracy at the completion of pre-training on $\mathcal{D}_\mathcal{T}$, calculated as, $\mathcal{F} = \frac{1}{\mathcal{T}-1} \sum_{i=1}^{\mathcal{T}-1} \max_{t \in \{1,\dots,\mathcal{T}-1\}} (acc_{t,i} - acc_{\mathcal{T},i})$.

Table 2.2: **Efficiency analysis.** GPU memory occupancy (GPU M.) is measured in GB. Throughput (T.P.) is measured in sample/sec. Both are estimated in single V100 with a batch size of 15 for STELLA++ and 9 for others.

Method	A→V		V→A		GPU M.↓	T.P.↑
	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$		
Finetune	3.63	10.50	4.11	10.12	18.34	**29.46**
ER	11.18	7.68	10.33	8.06	30.95	17.70
MIR	11.45	7.55	10.68	8.49	31.17	5.73
DER++	12.52	5.68	11.91	6.66	30.95	17.79
GMED	11.71	5.30	10.93	6.21	32.03	5.63
CLS-ER	12.74	6.30	12.27	6.58	32.50	15.24
LUMP	10.88	6.32	10.39	6.66	18.36	26.67
ESMER	13.58	5.80	13.33	5.79	31.45	14.88
STELLA (Ours)	14.16	4.38	14.07	4.65	17.45	17.29
STELLA+ (Ours)	15.44	4.62	15.26	3.89	**17.15**	18.11
STELLA++ (Ours)	**17.01**	**3.20**	**16.62**	**3.27**	24.69	-

Table 2.3: **Sampling methods.** Experiments with various sampling methods. LPIS: Localized Patch Importance Scoring in Section 2.4.1, RCA: Replay-guided Correlation Assessment Section 2.4.2.

Method	LPIS	RCA	A→V		V→A	
			$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
Random	–	–	12.64	6.46	12.55	6.58
MATS	–	–	12.91	6.55	12.70	6.80
STELLA (Ours)	–	–	12.52	5.68	11.91	6.66
	✓	–	13.44	5.50	13.27	5.94
	–	✓	13.40	5.30	12.94	5.44
	✓	✓	**14.16**	**4.38**	**14.07**	**4.65**

2.5.2 Analysis for Continual Audio-Video Pre-training

STELLA achieves superior Zero-shot Audiovisual Retrieval performance compared to strong baselines. We perform audio-to-video and video-to-audio zero-shot retrieval tasks in *Continual-VS* and *Continual-AS* to quantitatively assess the learned audio-video correlation from the continual pre-training (Table 2.1). For the *Continual-VS*, both *STELLA* and *STELLA+* outperform other baselines, exhibiting substantial enhancements of 0.58%p, 1.86%p and 0.74%p, 2.20%p in average audio-to-video and video-to-audio retrieval scores, respectively. In the *Continual-AS*, *STELLA+* exhibits prominent performance advantages, with 2.59%p and 3.69%p improvements in average audio-to-video and video-to-audio retrieval scores. Notably, our methods consistently achieve high R@1 scores across all tasks. These results imply that our approach of continually pre-training on the selected patches enhances the model's ability to comprehend the audio-video relationship by accurately capturing sparse spatio-temporal correlations. For a thorough investigation, we conduct further experiments with shuffled task orders in Section 2.12. We also explore the influence of rehearsal memory size on zero-shot task performances, presenting the results in Figure 2.6. Our methods consistently surpass other baselines, underscoring their effectiveness in adapting to diverse memory constraints.

STELLA is significantly efficient in terms of GPU Memory Consumption and Throughput. Pre-training on the spatio-temporally aligned subset of audio-video patches also enhances efficiency. In Table 2.2, we compare GPU memory occupancy and throughput across different methods. *STELLA* consumes significantly less GPU memory than baselines, even surpassing *Finetune* in efficiency. Compared to *DER++*, *STELLA+* achieves a 44.59% gain in efficiency, further enhancing throughput. In order to explore the benefits of reduced GPU memory usage, we conduct experiments with *STELLA+* with an increased batch size. Specifically, we increase the batch size by 1.66 times and denote this version of *STELLA+* as *STELLA++*. As shown in Table 2.2, *STELLA++* outperforms all baselines, including *STELLA+*. We expect that increasing batch size for contrastive learning-based models enhances the model's ability to accurately distinguish between various inputs and increases stability during continual pre-training. In the case of rehearsal memory burden, the extra cost required in *STELLA* for storing the queries, importance scores, and correlation scores in the memory is negligible (+ 0.16 GB), based upon the fact that the size of the memory itself is 5.47 GB and that *CLS-ER* and *ESMER* maintain additional models, which require + 1.42 GB and + 0.71 GB additional memory, respectively.

Core components in STELLA contribute to improving evaluation performance. To validate our patch selection method, we compare our two core components with *MATS* [51], an adaptive patch

Method	A→V			V→A		
	R@1	R@5	R@10	R@1	R@5	R@10
Finetune	1.00	4.15	6.44	1.33	3.19	6.15
ER	2.26	7.89	13.38	2.26	8.78	13.42
MIR	2.48	7.59	11.89	1.85	7.37	11.81
DER++	1.93	8.23	13.75	2.52	8.30	13.42
GMED	1.67	6.81	11.81	1.44	6.04	11.59
CLS-ER	2.15	8.45	12.93	2.15	7.63	12.82
LUMP	1.78	7.70	12.07	1.59	7.04	11.81
ESMER	2.33	8.37	13.78	2.30	8.48	13.93
STELLA (Ours)	**2.70**	8.70	13.96	**2.67**	8.81	14.30
STELLA+ (Ours)	2.37	**9.11**	**15.07**	2.44	10.14	**15.62**

(a) MSR-VTT audiovisual retrieval

Method	Accuracy
Finetune	57.04
ER	57.09
MIR	56.82
DER++	57.23
GMED	57.34
CLS-ER	57.23
LUMP	57.70
ESMER	57.72
STELLA (Ours)	58.20
STELLA—+ (Ours)	**58.54**
Multitask	59.94

(b) Audiovisual classification

Method	MIoU
Finetune	54.77
ER	54.64
MIR	54.69
DER++	55.42
GMED	55.92
CLS-ER	55.89
LUMP	55.34
ESMER	55.84
STELLA (Ours)	56.59
STELLA—+ (Ours)	**57.26**
Multitask	58.51

(c) Audiovisual segmentation

Figure 2.5: **Audiovisual downstream tasks.** We finetune models continually pre-trained on *Continual-VS* tasks. **(a):** Finetuning with the MSR-VTT [1] train dataset, we measure audiovisual retrieval performance. **(b):** We randomly initialize and finetune a MLP classifier, attached on the top of the models, using the entire *Continual-VS* dataset. **(c):** We finetune a randomly initialized decoder with the AVSBench [2] training dataset. MIoU (Mean Intersection over Union) measures the average overlap between predicted segments and ground truth segments. The best and the second best results are highlighted in **bold** and underline, respectively.

selection method aiming to discard redundant patches during video pre-training, and with a simple random patch selection method, denoted as *Random*. We decompose *STELLA* into Localized Patch Importance Scoring (*LPIS*) and Replay-guided Correlation Assessment (*RCA*). All the above methods follow the default sampling ratio and were built upon *DER++*. In *Continual-VS* zero-shot retrieval tasks, *LPIS* and *RCA* show competitive results against baselines including *MATS* and *Random* (Table 2.3). *LPIS* enhances the model's audio-video semantics comprehension. Conversely, *RCA* demonstrates more robustness in forgetting but with a lower average retrieval score, indicating a need for improved guidance in understanding audio-video semantics. Combining both components, *STELLA* achieves improved performances, emphasizing the importance of considering both the sparse correlation and forgetting in continual audio-video pre-training.

STELLA excels in various audiovisual downstream tasks. To evaluate the acquired transferable knowledge through continual audio-video pre-training, we perform diverse audiovisual downstream tasks. Compared to the earlier zero-shot retrieval tasks, we use the models that have been continually pre-trained up to the final task of *Continaul-VS*, and then evaluate them on different audiovisual datasets. First, we conduct audiovisual retrieval experiments on the MSR-VTT [1] dataset. We train the pre-trained models on the MSR-VTT training dataset according to the training objective in Section 2.3.1 and evaluate them on the MSR-VTT test dataset. As shown in Table 2.5 (a), our methods consistently outperform the baselines, demonstrating that our methods excel at understanding relationships in audio-video pairs. Second, we perform audiovisual classification experiments on the entire *Continual-VS* datasets with class labels. Specifically, we finetune a randomly initialized MLP classifier, which is attached to the top of the continually pre-trained models, using the datasets. This setup ensures that the classification results reflect the quality of audio-video representations learned throughout the continual audio-video pre-training process. Experimental results in Table 2.5 (b) demonstrate that our methods yield superior audio-video representations, leading to enhanced classification performance over baseline methods. This improvement is due to our approach's ability to identify patches with high audio-video correlation, thereby enhancing the model's comprehension of audio-video data during continual pre-training. Furthermore, we conduct audiovisual segmentation experiments. Following the experiments in [22], we finetune a randomly initialized decoder for the audiovisual segmentation task with the training dataset of the AVSBench [2]. The results, shown in Table 2.5 (c), indicate that our methods surpass the baselines. This suggests that our pre-trained models have a superior multimodal ability to

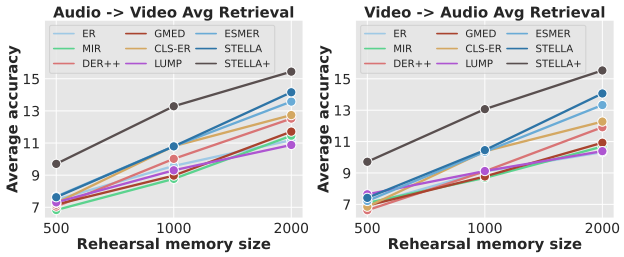


Figure 2.6: **Downstream performance on various rehearsal memory sizes.** We evaluate downstream task performances on the pre-trained models with various rehearsal memory sizes on the *Continual-VS*.

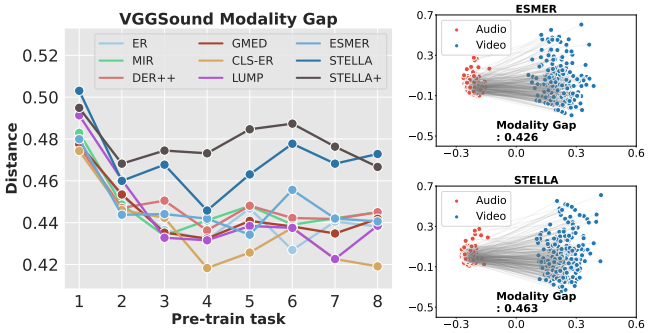


Figure 2.7: **Modality gap estimation. (Left)**: Estimation of modality gap after the completion of each task. (*Continual-VS*) **(Right)**: Visualizations of modality gap corresponding to the music task with the model pre-trained up to the last task in the *Continual-VS* dataset with *ESMER* (top) and our method (bottom).

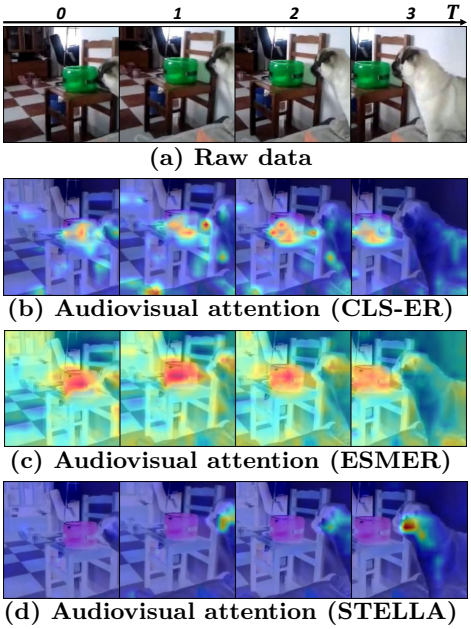


Figure 2.8: **Sound source localization (a)** A raw data describing a dog barking. **(b)˜(c)**: We visualize cross-attention maps using cosine similarity between each video patch and averaged audio embedding of the corresponding audio. **(d)**: We use the AVM module in *STELLA* to visualize cross-attention maps. For more examples, please see Figure 2.12.

spatially localize sound sources given corresponding audio, demonstrating the efficacy of our continual pre-training approach. Finally, we perform a sound source localization task on the AVE [52] dataset to assess the model's ability to detect sound sources within visual scenes. As shown in Figure 2.8, given audio containing a barking dog, all methods struggle to precisely locate the sound source, concentrating on the uncorrelated object (green bottle) in the visual scene. In contrast, the AVM module in *STELLA* stands out by precisely identifying the correct sound source, proving its efficacy in aligning multimodal data even in continual pre-training scenarios. This qualitative result further strengthens our quantitative evaluation of the audiovisual segmentation task in Figure 2.8. Additional results for other audiovisual downstream tasks, including event localization and retrieval tasks, are available in Section 2.12.

STELLA can preserve the modality gap between audio and video embeddings even after continual learning. Recent research in multimodal learning [53] reveals that embeddings cluster by modality in representation space. Such modality-dependent clustering behavior introduces the concept of modality gap, which refers to the distance between these clusters (Figure 2.7 (Right)). A larger modality gap is generally considered favorable under well-separated modality clusters since it indicates that the model can distinguish between different modalities effectively. Hence, in the context of continual audio-video pre-training, maintaining a large modality gap between the two modalities throughout tasks is desirable, as deviating from it suggests a departure from the optimal state. Hence, during continual pre-training, we estimate the modality gap at the end of each task, utilizing evaluation data of each task. The estimated modality gaps of baselines are presented in Figure 2.7 (Left). Our methods consistently maintain the highest modality gap compared to other approaches. Moreover, our methods exhibit small

modality gap declines, indicating that the models suffer less from the forgetting of previous multimodal correlations, which supports the validity of our approach in preventing *modality correlation overwriting* in Section 2.4.2 to address the issue of audio-video relation forgetting. Section 2.14 provides more analysis using the modality gap including *Continual-AS* and about two key components of our approach. Besides, some previous works [54] observe that reducing modality gaps also has benefits. Based on the modality gap analysis [54], there exists a modality gap that yields the best downstream task performances. However, we would like to emphasize that we use the modality to estimate the change in the modality gap throughout continual pre-training, not to find the best modality gap of the backbone model.

2.6 Conclusion

In this paper, we investigate the critical challenges in continual audio-video pre-training under the task-free scenario, where the model continuously learns a course of audio-video multimodal tasks sequentially and cannot access previous tasks and task oracle both on pre-training and fine-tuning. We empirically observe that the audio-video models suffer from the issue of sparse spatiotemporal correlation and representational forgetting of audio-video relationships. To overcome these limitations, we propose a novel continual audio-video multimodal pre-training method for the first time that adaptively captures sparse audio-video attention to learn accurate audio-video relationships while mitigating forgetting from previously learned relationships without requiring task identification.

2.7 Appendix

Organization The supplementary file is organized as follows: First, we explain the implementation details for our experiments in Section 2.8. Then, we outline the evaluation protocol of our experiments in Section 2.9. In Section 2.10, we elaborate on the audio-video self-supervised objectives used for pre-training the model. Additionally, Section 2.11 presents a detailed account of the training procedure for the AVM module. We provide additional experimental results in Section 2.12. Section 2.13 showcases the outcomes of our hyperparameter tuning process. Furthermore, in Section 2.14, we conduct more analysis on our experimental results using the modality gap. We present PyTorch-like pseudo code for audio patch selection in Section 2.15. We provide STELLA and STELLA+ algorithms in Section 2.16. Finally, in Section 2.17 we provide more examples of visualization that show challenges in audio-video lifelong pre-training.

2.8 Implementation Details

Hyperparameter configurations. We referred to the original papers for initial settings of hyperparameters of continual learning methods. Based on the initial settings, we tune the hyperparameters for our continual audio-video representation learning. Searched hyperparameters are listed in Table 2.4. In our method, α denotes a multiplier for the penalty loss to minimize the distance between obtained logits from the buffer instances and their logits stored at the past timestep. We also listed our pre-training and fine-tuning hyperparameters in Table 2.5.

Baselines. ER [41] employs rehearsal memory and learns the past data in the memory during training on the current task to mitigate forgetting. All the baselines below employ the rehearsal memory to store

Table 2.4: **Continual learning method hyperparameters.**

METHOD	Continual-VS	Continual-AS
ER	-	-
MIR	C : 5	C : 5
DER++	α : 0.5	α : 1.0
GMED	α : 0.1 β : 0.05 γ : 1.0	α : 0.1 β : 0.01 γ : 1.0
CLS-ER	λ : 0.1 α_S : 0.999 α_P : 0.999 r_S : 0.6 r_P : 0.8	λ : 0.1 α_S : 0.999 α_P : 0.999 r_S : 0.6 r_P : 0.8
LUMP	λ : 0.1	λ : 0.05
ESMER	α_l : 0.99 β : 1.0 γ : 0.15 α : 0.999 r : 0.2	α_l : 0.99 β : 1.0 γ : 0.2 α : 0.999 r : 0.2
STELLA (Ours)	α : 0.5 β : 0.4 ρ_a : 0.5 ρ_v : 0.5	α : 0.5 β : 0.1 ρ_a : 0.5 ρ_v : 0.5

Table 2.5: **Audio-Video pre-training and fine-tuning hyperparameters.**

	Pretrain		Finetune			
Dataset	Continual-VS	Continual-AS	MSR-VTT	AVC	AVS	AVE
Optimizer	Adam			AdamW		
Optimizer momentum	$\beta_1, \beta_2 = 0.95, 0.999$					
Learning rate	1e-4			1e-4	5e-4	1e-3
Weight decay	5e-7			5e-6		
Learning rate schedule	-		CosineScheduler			
Warmup epochs	-			3	2	
Epoch	10	15	15	10	20	15
Batch size	48	36	48	12		
GPUs	4 A100 or 4 V100		4 Titan X Pascal			
Audio Random Time Shifting	yes		no			
Audio Random Noise	yes		no			
Audio Norm Mean	-5.081					
Audio Norm STD	4.485					
Video MultiScaleCrop	yes					
Video Norm Mean	[0.485, 0.456, 0.406]					
Video Norm STD	[0.229, 0.224, 0.225]					

the subset of past data. MIR [47] introduces a strategy that retrieves data the model is likely to forget during the current task and trains the model with the retrieved data. To retrieve the data, it pseudo-updates the model with the data in the current step and finds the mini-batch of past data that gives the highest training loss. DER++ [42] matches stored logits in the rehearsal memory from past tasks with the current ones, ensuring a smoother transition and preventing abrupt changes in the logits during training. In our setting, we store both audio and video logits in the rehearsal memory and apply the method independently. GMED [48] tackles forgetting by using gradient information to update past data in the rehearsal memory. The data is updated to maximize interference of the current task to help the model retain past knowledge. Hence, it virtually updates the model with data from the current step and calculates the relative gradient by the past data to update the past data. CLS-ER [49] draws inspiration from the complementary learning system theory and maintains two models to retain short-term memories and long-term memories; one quickly adapts to new tasks and the other is slowly updated to retrain past knowledge. The slowly updated model transfers retained knowledge to the adaptable one, ensuring the retention of past information. LUMP [34] integrates past and current data by mixing the two data, rather than replaying the past data together with data from the current task to handle the forgetting issue. In our setting, we integrate the past and current video and audio respectively with the same ratio. Lastly, ESMER [44] employs a semantic memory model that has the same structure as the pre-trained model to slowly integrate the knowledge encoded in the weights. It refers to the memory model to alleviate the effect of the data from the current batch that induces abrupt drift in the learned representations in order to reduce forgetting. The suggested method effectively handles the abrupt representation changes

when the data distribution shifts.

2.9 Continual pre-training evaluation protocol

Audiovisual Dataset Configuration In this section, we specify how we design our continual audio-video pre-training experiments using two benchmark datasets: VGGSound and AudioSet. To mimic the data distribution shift due to the new audio-video semantics described in Section 2.1, we split the dataset according to the high-level categories. For the VGGSound dataset, we split the dataset into eight tasks based on the category labels [45]. Each task dataset consists of 6k-8k video clips from 20 different classes. We name it as *Continual-VS*. Then, we construct another pre-training dataset by combining the unused training dataset in VGGSound with the AudioSet-20k [46], resulting in a total of 104k video clips. We took care to exclude the unused VGGSound video samples whose class labels are present in the *Continual-VS*. Using the merged dataset, we pre-train the backbone weights before continual pre-training. This ensures that the model does not underperform at the initial continual pre-training stages while the model does not acquire any task-specific knowledge at the beginning. For the *Continual-VS* continual pre-training, we follow the task sequence: sports→music→vehicle→people→animals→home&nature→others part1(tools&others)→others part2(remaining others).

Similarly, we divided the AudioSet dataset into seven tasks, following class hierarchy information [46]. We name it as *Continual-AS*. Compared to *Continual-VS*, it exhibits imbalanced dataset size among tasks and contains much larger clips. To ensure proper pre-training for the *Continual-AS* experiments, we pre-train the model with the entire VGGSound dataset to avoid any potential performance issues during the initial stages of continual pre-training. We randomly shuffle the pre-train order and follow the task sequence: human→vehicle→nature→animal→others→home→music.

For downstream tasks, we use two audiovisual datasets: MSR-VTT [1] and AVE [52]. MRS-VTT consists of 10,000 video clips from 20 different categories. We collect video clips that contain audio modality on both the training dataset and the test dataset. This yields $\sim 6k$ and $\sim 0.9k$ video clips, respectively. We finetune the continually pre-trained models on the MSR-VTT training dataset and evaluate on the test dataset to perform audiovisual bi-directional retrieval tasks. In the case of the AVE dataset, it contains 4143 videos with 28 different event categories. Since the dataset is a subset of AudioSet, we conduct experiments on the pre-trained models on *Continaul-VS* only. With this dataset, we perform two downstream tasks: sound source localization, which requires the models to locate the sounding objects in the visual scene, and audiovisual event localization, which asks the model to classify audiovisual events for each time step. Given that all the downstream task datasets represent unseen data for the pre-trained models, they allow us to gauge the extent to which the model has acquired general knowledge of audio-video correlations during continual audio-video pre-training.

Audiovisual downstream task configuration When constructing audiovisual zero-shot retrieval tasks for model performance evaluation, we refer to the CAV [19] for both the *Continual-VS* and *Continual-AS* experiments. We employ the zero-shot retrieval task in CAV, but exclude evaluation samples that belong to the classes that are not included in any of the tasks. In the audiovisual event localization task, we follow experimental setups in [22]. In the fine-tuning stage of the retrieval and event localization task, we freeze the backbone model, connect it to a randomly initialized trainable linear classifier, and train the classifier with the training dataset to evaluate the acquired representation.

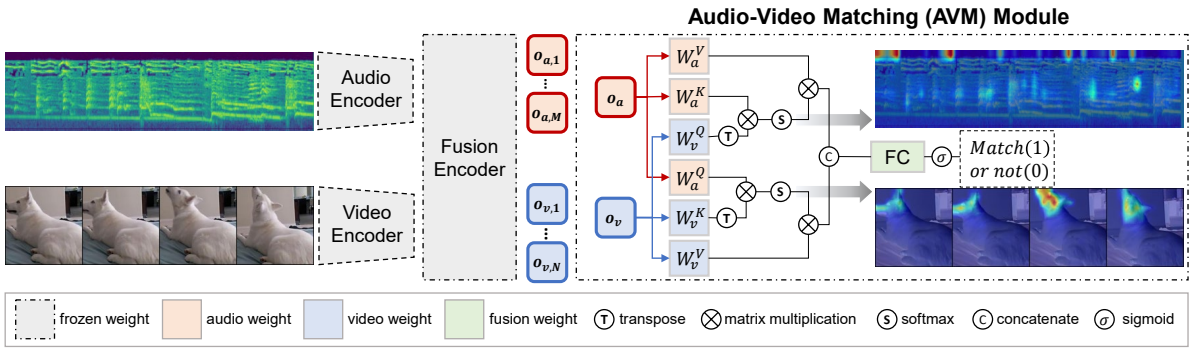


Figure 2.9: **Overview of AVM module**: The AVM (Audio-Visual Matching) module is self-supervised with the audio-video matching objective. It classifies if the given audio-video pair is positive(audio and video are from the same video) or negative(audio and video are from different videos).

2.10 Audio-Video Self-supervised objectives

Given audio-video data (X_a, X_v), we obtain D-dimensional embedding patches $\boldsymbol{a}$ and $\boldsymbol{v}$ as follows:

$$\boldsymbol{a} = \texttt{Conv2d}\left(X_a, \boldsymbol{w}_a\right), \quad \boldsymbol{v} = \texttt{Conv2d}\left(X_v, \boldsymbol{w}_v\right), \tag{2.7}$$

where $\boldsymbol{w}_a, \boldsymbol{w}_v$ denote the weights of convolutional layers, $\boldsymbol{a} \in \mathbb{R}^{B\times M\times D}$, and $\boldsymbol{v} \in \mathbb{R}^{B\times N\times D}$.

The backbone Transformer consists of an audio encoder $(E_a(\cdot))$, a video encoder $(E_v(\cdot))$, a multi-modal fusion encoder $(E_f(\cdot))$, and a decoder $(D(\cdot))$. Then we pre-train the model by minimizing the mask reconstruction loss ℓ^r:

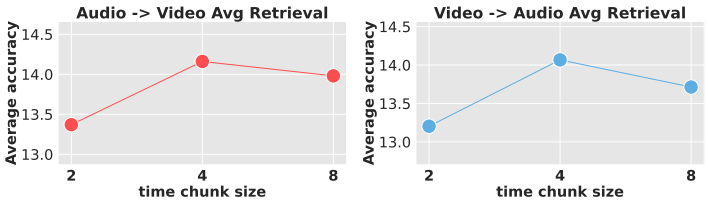
$$\begin{aligned}
&\tilde{\boldsymbol{a}},\ \tilde{\boldsymbol{v}} = E_f\left(E_a\left(\boldsymbol{m}_a \otimes \boldsymbol{a}\right),\ E_v\left(\boldsymbol{m}_v \otimes \boldsymbol{v}\right)\right), \\
&\ell^r = \ell^r_a + \ell^r_v = \frac{1}{B}\sum_{i=1}^{B}\left[\frac{\left(D\left(\tilde{\boldsymbol{a}}_i\right) - \boldsymbol{m}_{a,i} \otimes X_{a,i}\right)^2}{\left|\boldsymbol{m}_{a,i}\right|} + \frac{\left(D\left(\tilde{\boldsymbol{v}}_i\right) - \boldsymbol{m}_{v,i} \otimes X_{v,i}\right)^2}{\left|\boldsymbol{m}_{v,i}\right|}\right].
\end{aligned} \tag{2.8}$$

where $\otimes$ denotes vector-matrix multiplication while preserving the input's dimensionality. Random audio $\boldsymbol{m}_a$ and video mask $\boldsymbol{m}_v$ are drawn by a binary distribution. In this paper, we set a probability of 0.8 for masking, consistent with [21]. Using the unmasked patches, we aim to learn the model to reconstruct the masked audio and video patches.

In addition, we also minimize masked contrastive loss to learn the semantic relationship between audio and video representation pairs by pulling those that share the same semantics while pushing the others. Following by [19], we pass the masked input patches to audio and video encoders, and subsequently map obtained features (i.e., outputs) to the fusion encoder with modality-specific layer normalization for the masked contrastive learning:

$$\begin{aligned}
&\boldsymbol{c}_a = \texttt{MeanPool}\left(E_f\left(E_a\left(\boldsymbol{m}_a \otimes \boldsymbol{a}\right), LN_a\right)\right), \quad \boldsymbol{c}_v = \texttt{MeanPool}\left(E_f\left(E_v\left(\boldsymbol{m}_v \otimes \boldsymbol{v}\right), LN_v\right)\right), \\
&\ell^c = -\frac{1}{B}\sum_{i=1}^{B}\left[\texttt{log}\left(\frac{\texttt{exp}(\boldsymbol{c}_{a,i}^\top \boldsymbol{c}_{v,i}/\tau)}{\sum_{j=1}^{B}\texttt{exp}(\boldsymbol{c}_{a,i}^\top \boldsymbol{c}_{v,j}/\tau)}\right) + \texttt{log}\left(\frac{\texttt{exp}(\boldsymbol{c}_{v,i}^\top \boldsymbol{c}_{a,i}/\tau)}{\sum_{j=1}^{B}\texttt{exp}(\boldsymbol{c}_{v,i}^\top \boldsymbol{c}_{a,j}/\tau)}\right)\right],
\end{aligned} \tag{2.9}$$

where τ is temperature hyperparameter, and LN_a and LN_v indicate modality-specific layer normalization for audio and video each.



(a) Time chunk sizes

Method	A→V		V→A	
	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
Frequency	13.42	5.51	12.76	6.40
No constraint	12.67	6.55	12.78	6.61
Time	**14.16**	**4.38**	**14.07**	**4.65**

(b) Audio selection methods

Figure 2.10: **Variation of audio patch selection. (a)**: Average retrieval task performance on various time chunk sizes. **(b)**: Average retrieval task performance on various audio selection methods.

2.11 Training of Audio-Video Matching module

AVM training procedure. In the following section, we describe the training process of the AVM module, as illustrated in Figure 2.9. Given audio-video patch pairs $(\boldsymbol{a}, \boldsymbol{v})$ with the batch size of B, we propagate patch inputs to the frozen encoder for each modality and obtain audio-video representation pairs. In order to update the module to capture the multimodal correlation between audio and its video pair, we randomly split them into positive and negative pairs, where we construct negative pairs by randomly shuffling the audio patches to pair with unmatched video patches. Next, we project the obtained positive and negative pairs into fusion space $(\boldsymbol{o}_a,\ \boldsymbol{o}_v) = E_f(E_a(\boldsymbol{a}), E_v(\boldsymbol{v}))$ through the fusion encoder. Subsequently, the input pairs are fed into the AVM module. They are projected to keys, queries, and values for the cross-attention operation, by passing through trainable projection layers. The above process can be summarized as follows:

$$\begin{aligned} \boldsymbol{q}_a = \boldsymbol{o}_a \mathcal{W}_a^Q,\ \boldsymbol{k}_a = \boldsymbol{o}_a \mathcal{W}_a^K,\ \boldsymbol{v}_a = \boldsymbol{o}_a \mathcal{W}_a^V, \quad \boldsymbol{q}_v = \boldsymbol{o}_v \mathcal{W}_v^Q,\ \boldsymbol{k}_v = \boldsymbol{o}_v \mathcal{W}_v^K,\ \boldsymbol{v}_v = \boldsymbol{o}_v \mathcal{W}_v^V, \\ \boldsymbol{V}_a = \texttt{Softmax}\left(\mu(\boldsymbol{q}_v, \boldsymbol{k}_a, \beta{=}1)\right) \cdot \boldsymbol{v}_a, \quad \boldsymbol{V}_v = \texttt{Softmax}\left(\mu(\boldsymbol{q}_a, \boldsymbol{k}_v, \beta{=}1)\right) \cdot \boldsymbol{v}_v, \end{aligned} \tag{2.10}$$

where the projections $\mathcal{W}_a^Q, \mathcal{W}_a^K, \mathcal{W}_a^V, \mathcal{W}_v^Q, \mathcal{W}_v^K, \mathcal{W}_v^V \in \mathbb{R}^{D \times H \times d}$ are trainable parameter matrices; $D = H * d$. $\boldsymbol{V}_a \in \mathbb{R}^{B \times H \times N \times d}$, $\boldsymbol{V}_v \in \mathbb{R}^{B \times H \times M \times d}$ are values highlighted by the cross-attention maps.

Next, we average the values head-wise and patch-wise, and concatenate the resulting two values into $\boldsymbol{va} \in \mathbb{R}^{B \times 2D}$ in order to merge the multimodal information. Then it is passed to fully connected (`FC`) layers, which serve as the classification head. These `FC` layers take $\boldsymbol{va}$ as input, generating a vector $\hat{\boldsymbol{y}} \in \mathbb{R}^B$ that predicts whether each input pair corresponds to a negative of positive pair. For training the AVM module, we employ the binary cross-entropy loss to classify audio-video pairs, i.e.,

$$\begin{aligned} \widehat{\boldsymbol{V}}_{av} &= \texttt{Concat}\left(\texttt{MeanPool}\left(\boldsymbol{V}_a\right), \texttt{MeanPool}\left(\boldsymbol{V}_v\right)\right), \\ \widehat{\boldsymbol{y}} &= \texttt{Sigmoid}\left(\texttt{FC}(\widehat{\boldsymbol{V}}_{av})\right),\ \mathcal{L}^{avm} = -\boldsymbol{y}\left(\texttt{log}(\widehat{\boldsymbol{y}})\right), \end{aligned} \tag{2.11}$$

Here, $\boldsymbol{y} = \{0, 1\}^B$ represents ground truth labels, with $\boldsymbol{y}_i$ taking the value 0 when the *ith* input audio-video pair is a negative pair and 1 otherwise. We pre-train the AVM module along with the backbone model. During the weight update process in the AVM module, the gradient computed from the audio-video matching objective does not propagate through the backbone encoder. This design choice ensures exploiting the AVM at a low cost. Moreover, the AVM only increases 3.18% of the total backbone model size (707.8 MB), which is efficient compared to methods like *CLS-ER* or *ESMER* which require additional backbones during training.

Table 2.6: **Shuffle task orders.** Results of audiovisual zero-shot retrieval task on *Continual-VS* and *Continual-AS*. We randomly shuffle the task sequences for continual pre-training. For the *Continual-VS*, we follow the task order: music → others part1 → home&nature → sports → others part2 → vehicle → animals → people. For the *Continual-AS*, we follow the task order: nature → human → home → vehicle → music → animal → others. R@K means top-K recall. The results are the means of 3 independent runs. The best and the second best results are highlighted in **bold** and underline, respectively.

	Method	Continual-VS R@1		R@5		R@10		Avg		Continual-AS R@1		R@5		R@10		Avg	
		$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
Audio-to-Video	Finetune	0.80	4.15	2.96	12.23	5.05	16.91	2.94	11.10	1.50	4.72	5.49	10.41	9.80	11.91	5.60	9.01
	ER	3.89	3.06	12.10	6.55	18.30	7.74	11.43	5.78	4.52	3.16	12.72	6.93	18.83	8.00	12.02	6.03
	MIR	4.02	2.97	12.54	6.16	17.99	8.09	11.52	5.74	4.69	2.95	13.22	6.50	18.98	8.81	12.30	6.09
	DER++	4.23	3.35	12.92	7.31	18.62	9.45	11.92	6.70	4.32	4.27	12.29	8.46	18.74	10.18	11.78	7.64
	GMED	3.90	2.94	11.51	7.41	17.65	8.87	11.02	6.41	4.70	**2.48**	12.56	4.55	18.62	**5.05**	11.96	**4.03**
	CLS-ER	3.94	3.35	12.96	7.19	18.09	10.66	11.66	7.07	5.16	2.97	14.33	6.88	20.24	8.74	13.24	6.20
	LUMP	4.06	**2.18**	13.21	**4.66**	19.34	**5.58**	12.20	**4.14**	4.45	3.40	13.05	6.25	19.45	7.28	12.32	5.64
	ESMER	4.38	3.36	13.31	8.28	19.39	9.20	12.36	6.95	5.43	3.85	15.81	6.20	21.40	8.81	14.21	6.29
	STELLA (Ours)	4.72	2.89	14.17	5.74	19.94	5.74	12.94	4.79	4.97	3.47	13.91	5.59	20.30	6.70	13.06	5.25
	STELLA+ (Ours)	**4.90**	3.19	**16.42**	4.72	**23.49**	5.89	**14.94**	4.60	**5.77**	3.90	**17.51**	**4.49**	**23.72**	7.07	**15.67**	5.15
	Multitask	6.45	–	20.19	–	29.01	–	18.55	–	8.28	–	24.14	–	33.74	–	22.05	–
Video-to-Audio	Finetune	0.78	3.77	3.00	11.68	5.21	15.86	3.00	10.44	1.42	5.11	6.54	10.30	10.43	13.48	6.13	9.63
	ER	3.57	2.76	11.66	7.67	16.75	10.76	10.66	7.06	4.01	4.31	12.47	7.27	19.32	9.26	11.93	6.95
	MIR	3.35	3.15	11.37	7.74	16.62	10.11	10.45	7.00	4.25	3.43	12.92	6.93	19.43	9.78	12.20	6.71
	DER++	4.08	3.10	12.78	9.02	18.77	11.30	11.88	7.81	4.31	4.35	12.60	9.59	18.93	12.27	11.95	8.74
	GMED	3.42	3.80	11.45	7.76	17.06	9.94	10.64	7.17	4.20	**1.87**	12.97	**6.04**	19.98	**8.11**	12.38	**5.34**
	CLS-ER	3.49	3.85	12.28	8.05	17.75	11.31	11.17	7.74	4.85	5.48	13.37	9.17	19.69	11.36	12.64	8.67
	LUMP	3.98	**1.67**	12.44	**5.17**	18.11	**7.27**	11.51	**4.70**	4.23	4.06	13.53	6.09	19.27	9.53	12.34	6.56
	ESMER	4.44	3.35	13.32	8.69	19.47	10.27	12.41	7.44	5.12	5.48	14.73	8.79	20.35	12.41	13.40	8.89
	STELLA (Ours)	4.18	2.54	13.81	6.56	19.90	8.88	12.63	5.99	4.86	2.92	14.20	6.41	20.00	9.82	13.02	6.38
	STELLA+ (Ours)	**5.28**	1.81	**15.35**	6.33	**21.97**	8.01	**14.20**	5.38	**5.57**	3.80	**16.67**	6.96	**23.91**	9.28	**15.38**	6.68
	Multitask	6.85	–	21.93	–	30.63	–	19.80	–	8.05	–	25.81	–	35.60	–	23.15	–

2.12 Additional Experimental Results

Audio patch selection strategy. When executing the selection of audio patches guided by the audio importance score $\boldsymbol{I}_a$, our approach involves selecting patches in time-wise segments, following the procedure detailed in Algorithm 1. As spectrogram patches exhibit local correlation driven by their temporal continuity [55], the strategy for audio patch selection becomes pivotal in maintaining these intrinsic properties. The challenge lies in striking a balance between retaining time continuity and eliminating redundant information within the spectrogram.

In pursuit of this balance, we conduct various experiments on the audio patch selection approach. The width of the time chunk assumes significance; a chunk that is too narrow could disrupt time continuity, while one that is excessively wide might not concisely capture core information. To validate our approach and assess the efficacy of time-wise chunk selection, we conduct two distinct sets of experiments.

The first experiment involves evaluating the model's performance across varying time chunk widths. A noteworthy observation from Figure 2.10 (a): adopting a size of 2 results in a noticeable performance decline. This potentially signifies the criticality of upholding the local correlation inherent in audio patches. Moving on to the second experiment, we explore various selection methods, inspired by the spectrogram masking techniques detailed in [55]. We test two variants of audio patch selection: Frequency indicates an approach of choosing audio patches frequency-wise, while No-constraint indicates selecting audio patches without any constraints; applying the same patch selection procedure as in the video patch selection. As shown in Figure 2.10 (b), time-wise selection exhibits superior performance compared to alternative audio selection methodologies, meaning that preserving audio information in time-chunk minimizes loss of audio properties.

Shuffle task orders. In addition to the main experiment results presented in Table 2.1, we conduct supplementary investigations with the intention of enhancing the reliability of our findings. Specifically, we carry out experiments on shuffled task sequences. For the *Continual-VS*, we randomize the original pre-train task sequence, leading to modified order: music→others part1→home&nature→sports→others

Method	A→V			V→A		
	R@1	R@5	R@10	R@1	R@5	R@10
Finetune	0.52	2.81	4.82	0.67	2.82	5.08
ER	1.48	6.70	11.48	1.74	7.19	12.07
MIR	1.56	5.97	10.23	1.85	6.93	11.89
DER++	2.74	9.08	14.49	2.45	9.49	14.60
GMED	2.07	8.04	13.11	2.70	8.44	12.89
CLS-ER	2.78	<u>9.40</u>	14.43	<u>2.89</u>	8.73	14.54
LUMP	2.33	8.15	12.75	2.04	7.93	12.45
ESMER	<u>2.89</u>	9.70	<u>15.56</u>	2.70	**10.22**	<u>16.04</u>
STELLA (Ours)	2.74	9.26	15.37	2.85	9.48	15.56
STELLA+ (Ours)	**2.93**	**10.22**	**16.33**	**3.67**	**10.22**	**16.26**

(a) MSR-VTT audiovisual retrieval

	Method	Acc
AVE	Finetune	52.56
	ER	54.98
	MIR	56.13
	DER++	55.81
	GMED	55.98
	CLS-ER	<u>56.39</u>
	LUMP	55.06
	ESMER	55.60
	STELLA (Ours)	**56.68**
	STELLA+ (Ours)	**56.68**
	Multitask	57.73

(a) Audiovisual event localization

Figure 2.11: **Additional downstream tasks (a):** MSR-VTT audiovisual retrieval. MSR-VTT audiovisual retrieval task performances. We use the models continually pre-trained until completion of the last task of *Continual-AS*. **(b):** We randomly initialize and finetune a MLP classifier with AVE dataset [3]. The best and the second best results are highlighted in **bold** and <u>underline</u>, respectively.

part2→vehicle→animals→people. Likewise, in the case of the *Continual-AS* experiment, we apply a similar task sequence shuffling, resulting in the following order: nature→human→home→vehicle→music→animal→others. Note that the *Continual-VS* experiment is conducted on 36 batch size, unlike the main *Continual-VS* experiment which is conducted on 48 batch size. We present the corresponding audiovisual zero-shot retrieval task results in Table 2.6. Our method shows competitive or better performance compared to other baselines, which coincides with the results in Table 2.1. This indicates that our method is robust under varying conditions, thereby enhancing the credibility of our analysis.

MSR-VTT retrieval task. We provide additional experiment results on the MSR-VTT retrieval task in Figure 2.11 (a). In this experiment, we use the models continually pre-trained up to the last task of *Continual-AS*. We follow the training configurations in Table 2.5. The experiment results show that our methods consistently show competitive results, which supports that our methods obtain general audio-video correlations that are transferable to retrieval tasks.

Audiovisual event localization. We conduct an audiovisual event localization (AVE) task to showcase the effectiveness of our method in precisely aligning audio and video streams. Following the experimental setup outlined in [22], we utilize the AVE dataset [3] for the experiment. To assess whether continually pre-trained models can adapt to the downstream task involving the unseen dataset, we use the model pre-trained on all tasks in the sequence within the *Continual-VS* experiment. The training process adheres to the hyperparameters described in Table 2.5, wherein the backbone model remains frozen while training the linear classifier. We present the summarized result in Figure 2.11 (b). This result demonstrates that our method surpasses other baseline methods. This underscores the strength of our method in adapting the downstream task that necessitates a sophisticated grasp of audio-video alignment at a high level.

Sound source localization. We provide more visualization results of the sound source localization in Figure 2.12. Our method consistently shows superior ability in locating potential sound sources in the visual scenes.

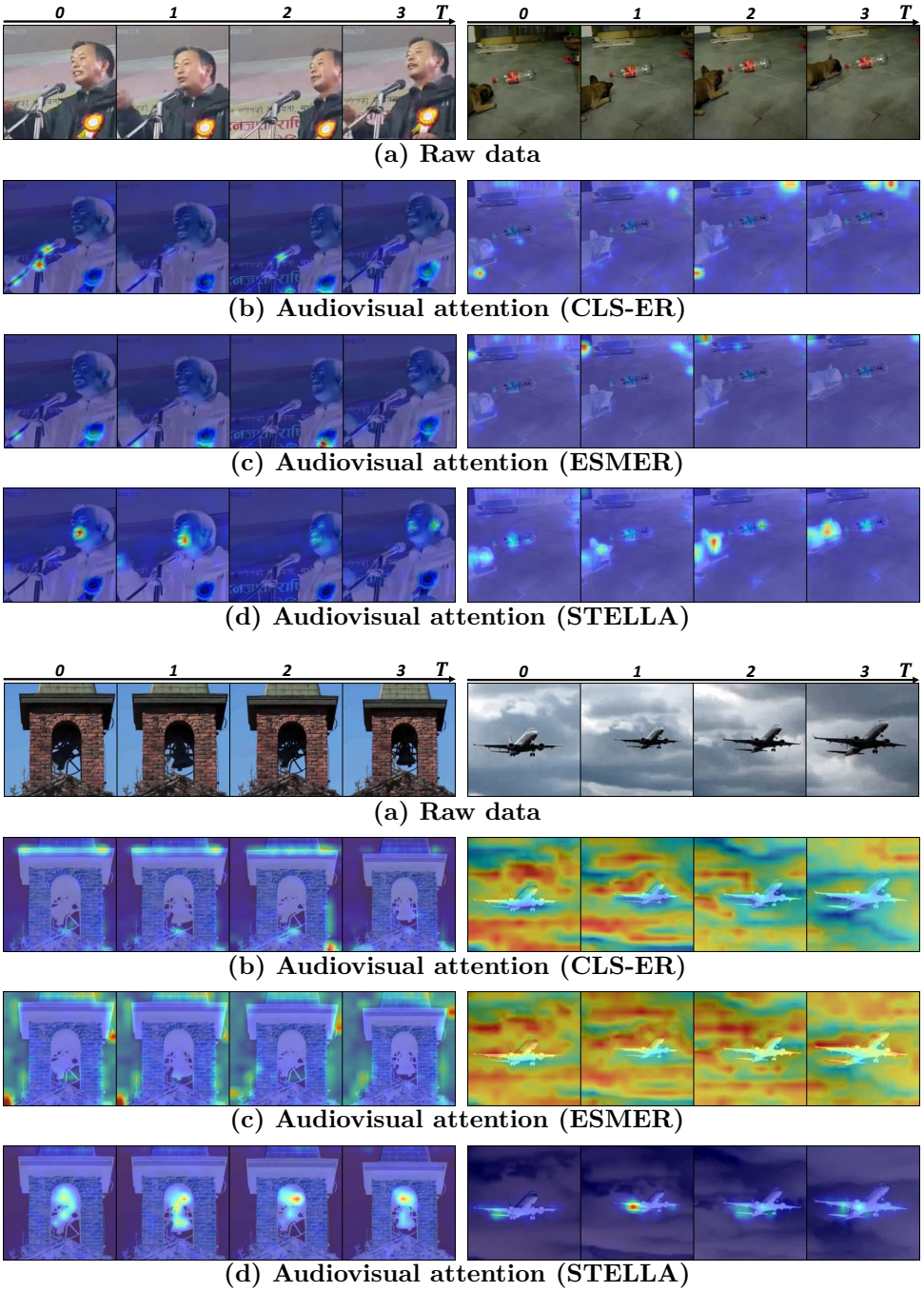
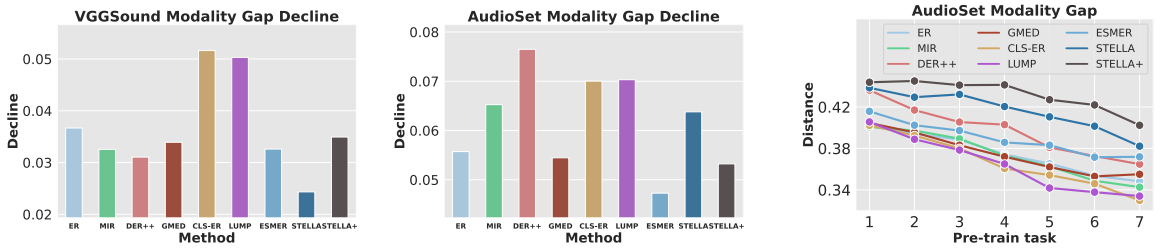


Figure 2.12: **Sound source localization (a)** Examples of raw video frames. **(b)˜(c)**: We visualize cross-attention maps using cosine similarity between each video patch and averaged audio embedding. **(d)**: We use the AVM module in *STELLA*, continually pre-trained with the backbone mode, to visualize cross-attention maps. Our method is much more effective in capturing potential sound sources compared to the ability of the backbone to capture the sources.



(a) Modality gap average decline **(b) Modality gap after each task**

Figure 2.13: **Modality gap estimation. (a)**: Average modality gap decline between the modality gap estimated at the completion of the last task and the modality gap estimated at the completion of each task. **(b)**: Estimation of modality gap after the completion of each task (*Continual-AS*).

2.13 Hyperparamter Tuning Results

Patch sampling ratio. Central to our approach is the identification of patches that exhibit a high localized alignment with their corresponding modality pairs while being robust to catastrophic forgetting of learned representation, enabling the retention of meaningful information. Achieving the right balance in the sampling ratio is critical: an excessively low sampling ratio hinders the model from accessing essential data, while an overly high ratio hampers the model's ability to disregard redundant or forget-inducing information.

Table 2.7: **Retrieval result by sampling ratios.**

	Ratio(%)	A→V		V→A	
		$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
ρ_a	37.5	13.76	4.77	13.52	5.53
	50	**14.16**	**4.38**	**14.07**	**4.65**
	62.5	13.77	5.04	13.46	5.06
ρ_v	37.5	13.35	5.57	13.39	5.93
	50	**14.16**	**4.38**	**14.07**	**4.65**
	62.5	13.82	4.50	13.53	5.27

For the audio sampling ratio, we systematically assess three options —37.5%, 50%, and 62.5%— while maintaining the video sampling ratio ρ_v at 50%. Table 2.7 shows that sampling 50% of audio patches ensures high performance compared to the other sampling ratios. It is noteworthy that the other sampling ratios still yield competitive performance compared to the baselines. As we transition to optimizing the sampling ratio for video patches, we conduct experiments using three sampling ratios -37.5%, 50%, and 62.5%- alongside the audio sampling ratio ρ_a at 50%. As demonstrated in Table 2.7, employing a 50% video sampling ratio ensures high performance.

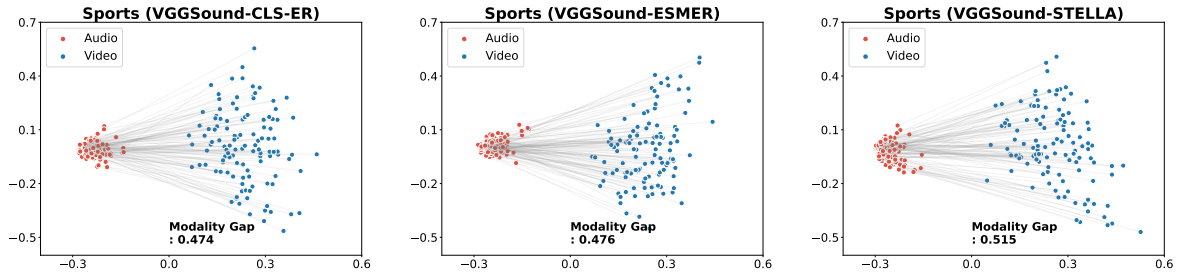
Inference temperature in AVM module. In our approach, we actively harness cross-attention maps from the AVM module computed in Equation2.1. During inference, we set the temperature hyperparameter β to 0.4 for the *Continual-VS* experiments. To examine the significance of β, we explore a range of the hyperparameter values, specifically 0.1, 0.4, and 0.5. The results, as summarized in Table 2.8, indicate that the optimal temperature values typically reside within the range of approximately 0.1 to 0.4. This suggests the need for heightened emphasis on discriminative audio and video patches in order that those patches are more frequently selected in our selection framework in Equation 2.5 and in Algorithm 1.

Table 2.8: **Retrieval result by temperature values.**

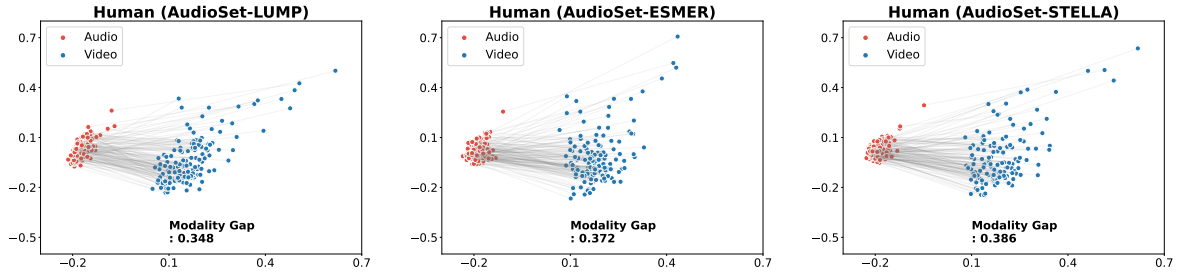
β	A→V		V→A	
	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$
0.1	13.91	5.42	**14.23**	4.97
0.4	**14.16**	**4.38**	14.07	**4.65**
0.5	13.37	5.27	13.50	5.84

2.14 Additional Analysis of Modality Gap

Comprehensive analysis In the main paper, we examine the performance improvements of our approach in the context of continual audio-video pre-training with respect to the modality gap. In this



(a) Visualization in representation space (Continual-VS, sports)



(b) Visualization in representation space (Continual-AS, human)

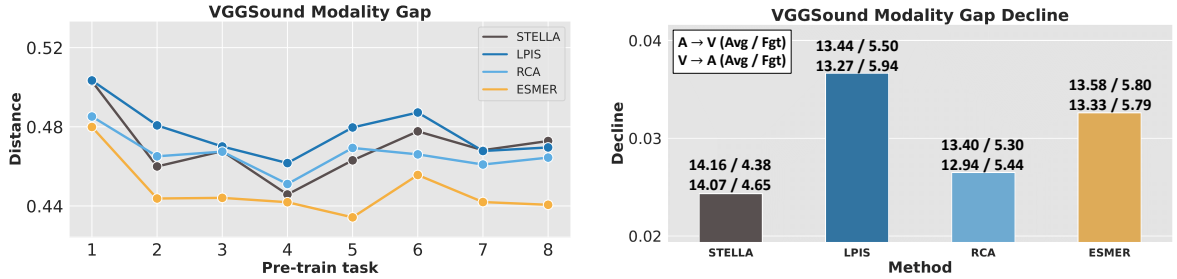
Figure 2.14: **Modality gap visualization. (a)**: Visualizations of the modality gap corresponding to the sports task with the model pre-trained up to the last task in the *Continual-VS* experiment. **(b)**: Visualization of the modality gap corresponding to the human task with the model pre-trained up to the last task in the *Continual-AS* experiment.

section, we conduct a more detailed analysis; covering differences in the modality gap (Figure 2.13 (a)), exploring the modality gap within the *Continual-AS* (Figure 2.13 (b)), and providing additional visualizations of the modality gap to support the effectiveness of our approach (Figure 2.13 (c)).

In Figure 2.13 (a), our approach stands out with the smallest average modality gap difference. However, our approach does not exhibit high resistance to modality gap fluctuations within the *Continual-AS* experiment. An interesting observation emerges when comparing the average modality gap difference with the average forgetting in Table 2.1; a smaller average modality gap difference seems to correspond to lower average forgetting in the zero-shot retrieval tasks. This aligns with the relatively high average forgetting of our approach in the *Continual-AS* experiment, suggesting that the modality gap difference holds potential as a metric for assessing the extent of forgetting in audio-video correlation. Meanwhile, our approach consistently maintains the highest modality gap in all pre-train tasks (Figure 2.13 (b)), which explains the high average accuracy of our approach in the *Continual-AS* retrieval tasks.

We take our analysis a step further by visually representing the modality gap. In Figure 2.14 (a), we visualize evaluation audio-video data pairs from the sports task in the *Continual-VS* experiments. Similarly, in Figure 2.14 (b), we visualize data from the human task in the *Continual-AS* experiments. In both visualizations, we use the models that completed the continual pre-training phase. Remarkably, our approach consistently yields a larger gap in both cases. This suggests that the modality gap established from the initial task (sports, human) is effectively maintained, enabling the models to distinguish between different modalities, ultimately leading to enhanced performance.

Analysis on STELLA components We estimate the modality gap of two key components within our proposed method: *ELPP* (Efficient Localized Patch Pooling Section 2.4.1) and *RCA* (Replay-guided Correlation Assessment Section 2.4.2). The *ELPP* consistently exhibits the highest modality gap across the tasks, as depicted in Figure 2.15 (a). This underscores the effectiveness of the proposed method in



(a) Modality gap after each task **(b) Modality gap average decline**

Figure 2.15: **Modality gap estimation for each component of our proposed method. (a)**: Estimation of modality gap after completing each task. **(b)**: Average decline in modality gap between the completion of the last task and the completion of each task.

Section 2.4.1 in identifying patches that demonstrate high localized alignment with their modality pairs. Consequently, the *ELPP* achieves better audio and video clustering within the multi-modal representation space, resulting in enhanced average accuracy in Table 2.3. This observation strongly supports our claim that the method outlined in Section 2.4.1 adeptly selects informative multi-modal patches from raw data.

The *RCA* illustrates a relatively minor modality gap difference, as indicated in Figure 2.15 (b). During the continual pre-training, the modality gap between the audio and video exhibits robustness to the effect of changing distribution. Hence, the model maintains learned audio-video alignment. This explains the small average forgetting exhibited by the *RCA* in Table 2.3. It affirms our claim that the method introduced in Section 2.4.2 proficiently selects forget-robust patches.

2.15 Audio Patch Selection Pseudo Code

Algorithm 1 Audio time chunk selection in a PyTorch-like Style.

```
# I_a:  audio patch importance score
# P_a:  audio pruning probability matrix
# L_c:  audio time chunk size
# kappa_a:  target number of audio tokens
# num_time:  the number of tokens in time dimension
# num_freq:  the number of tokens in frequency dimension
def audio_time_chunk_selection(I_a,P_a):
    F_a=bernoulli(P_a)
    F_a=F_a.reshape(num_time, num_freq)
    F_a_t=F_a.sum(dim=1) # # of pruned patches
    I_a_t=I_a.reshape(num_time, num_freq)
    I_a_t=I_a_time.sum(dim=1) # Time-wise importance
    I_a_c=avg_pool(I_a_t, kernel_size=L_c) # Chunk-wise importance
    num_chunk=len(I_a_c)
    t_select=multinomial(I_a_c, num_samples=num_chunk)
    num_tokens=0
    for j in range(num_chunk):
       t=t_select[j]
       num_prune=F_a_t[t*L_c:(t+1)*L_c].sum() # # of pruned patches
       num_tokens+=(L_c*num_freq - num_prune) # Count # of patches
       if num_tokens > kappa_a:
          F_last=F_a[t*L_c:(t+1)*L_c].view(-1)
          F_last_accum=cumsum(flip(~F_last))
          prune_tail_idx= F_last_accum == num_tokens-kappa_a
          F_last[-(prune_tail_idx+1):]=True # Prune tail of last chunk
          F_a[t*L_c:(t+1)*L_c]=F_last.reshape(num_time,num_freq)
          for k in range(j+1, num_chunk):
             t_prune=t_select[k]
             F_a[t_prune*L_c:(t_prune+1)*L_c]=True
          break
    F_a=F_a.view(-1).float()
    S_tilde_a=argsort(F_a) # Forget-robust audio sorted indices
    return S_tilde_a
```

2.16 Algorithms of STELLA and STELLA +

Algorithm 2 Continual Pre-training of STELLA

INPUT Dataset $\mathcal{D}_i$, model $f_{\theta,i-1}$, AVM module $h_{\Theta,i-1}$, rehearsal memory $\mathcal{M}$.
for batch $(X_a,\ X_v) \sim \mathcal{D}_i$ **do**
  $\boldsymbol{k}_a, \boldsymbol{q}_a, \boldsymbol{A}_a,\ \boldsymbol{k}_v, \boldsymbol{q}_v, \boldsymbol{A}_v \leftarrow \text{AVM}(X_a,\ X_v)$
  $\boldsymbol{I}_a,\ \boldsymbol{I}_v \leftarrow \textsc{Importance}(\boldsymbol{A}_a,\ \boldsymbol{A}_v)$
  $\widehat{\boldsymbol{k}}_a, \widehat{\boldsymbol{q}}_a,\ \widehat{\boldsymbol{k}}_v, \widehat{\boldsymbol{q}}_v \leftarrow \textsc{Sort}(\boldsymbol{k}_a, \boldsymbol{q}_a, \boldsymbol{I}_a, \boldsymbol{k}_v, \boldsymbol{q}_v, \boldsymbol{I}_v)$
  $X_a^p, X_v^p,\ \widehat{\boldsymbol{q}}_a^p, \widehat{\boldsymbol{q}}_v^p,\ \boldsymbol{I}_a^p, \boldsymbol{I}_v^p,\ \boldsymbol{C}_a^p, \boldsymbol{C}_v^p \leftarrow \mathcal{M}$
  $\boldsymbol{C}_a,\ \boldsymbol{C}_v \leftarrow \textsc{Compare}(\widehat{\boldsymbol{k}}_a, \widehat{\boldsymbol{q}}_v, \widehat{\boldsymbol{q}}_v^p,\ \widehat{\boldsymbol{k}}_v, \widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{q}}_a^p)$
  $\widehat{X}_a, \widehat{X}_a^p \leftarrow \textsc{Pick}([X_a, X_a^p], [\boldsymbol{I}_a, \boldsymbol{I}_a^p], [\boldsymbol{C}_a, \boldsymbol{C}_a^p])$
  $\widehat{X}_v, \widehat{X}_v^p \leftarrow \textsc{Pick}([X_v, X_v^p], [\boldsymbol{I}_v, \boldsymbol{I}_v^p], [\boldsymbol{C}_v, \boldsymbol{C}_v^p])$
  $\mathcal{M} \leftarrow \mathcal{M} \cup (X_a, X_v,\ \widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{q}}_v,\ \boldsymbol{I}_a, \boldsymbol{I}_v,\ \boldsymbol{C}_a, \boldsymbol{C}_v)$
  $\Theta \leftarrow \Theta - \eta \nabla h_{\Theta,i-1}\,(X_a,\ X_v)$
  $\theta \leftarrow \theta - \eta \nabla f_{\theta,i-1}\left([\widehat{X}_a, \widehat{X}_a^p],\ [\widehat{X}_v, \widehat{X}_v^p]\right)$
end for

Algorithm 3 Continual Pre-training of STELLA+

INPUT Dataset $\mathcal{D}_i$, model $f_{\theta,i-1}$, AVM module $h_{\Theta,i-1}$, rehearsal memory $\mathcal{M}$.
for batch $(X_a,\ X_v) \sim \mathcal{D}_i$ **do**
  $\boldsymbol{k}_a, \boldsymbol{q}_a, \boldsymbol{A}_a,\ \boldsymbol{k}_v, \boldsymbol{q}_v, \boldsymbol{A}_v \leftarrow \text{AVM}(X_a,\ X_v)$
  $\boldsymbol{I}_a,\ \boldsymbol{I}_v \leftarrow \textsc{Importance}(\boldsymbol{A}_a,\ \boldsymbol{A}_v)$
  $\widehat{\boldsymbol{k}}_a, \widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{k}}_v, \widehat{\boldsymbol{q}}_v \leftarrow \textsc{Sort}(\boldsymbol{k}_a, \boldsymbol{q}_a, \boldsymbol{I}_a,\ \boldsymbol{k}_v, \boldsymbol{q}_v, \boldsymbol{I}_v)$
  $\widehat{X}_a^p, \widehat{X}_v^p,\ \widehat{\boldsymbol{q}}_a^p, \widehat{\boldsymbol{q}}_v^p \leftarrow \mathcal{M}$
  $\boldsymbol{C}_a,\ \boldsymbol{C}_v \leftarrow \textsc{Compare}(\widehat{\boldsymbol{k}}_a, \widehat{\boldsymbol{q}}_v, \widehat{\boldsymbol{q}}_v^p,\ \widehat{\boldsymbol{k}}_v, \widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{q}}_a^p)$
  $\widehat{X}_a \leftarrow \textsc{Pick}(X_a, \boldsymbol{I}_a, \boldsymbol{C}_a,)$
  $\widehat{X}_v \leftarrow \textsc{Pick}(X_v, \boldsymbol{I}_v, \boldsymbol{C}_v)$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \left(\widehat{X}_a, \widehat{X}_v, \widehat{\boldsymbol{q}}_a, \widehat{\boldsymbol{q}}_v\right)$
  $\Theta \leftarrow \Theta - \eta \nabla h_{\Theta,i-1}\,(X_a,\ X_v)$
  $\theta \leftarrow \theta - \eta \nabla f_{\theta,i-1}\left([\widehat{X}_a, \widehat{X}_a^p],\ [\widehat{X}_v, \widehat{X}_v^p]\right)$
end for

2.17 Visualization of Fading Audio-Visual Attention

As shown in Figure 2.3 of the main paper, we tackle the problem of forgetting past audio-video correlation by visualizing the attention maps. In Figure 2.16, we provide additional examples that vividly illustrate the challenge of forgetting past correlation as the model undergoes pre-training on sequential tasks.

In the top-left example of Figure 2.16, we observe a video example where a person is engaged in rope skipping. The initial attention map concentrated on the feet (b). However, as the model adapts to new tasks, the attention map is shifted solely to the person's face (c), implying the gradual erosion of the correlation between the sound of rope skipping and the corresponding jumping motion. In the top-right example of Figure 2.16, the attention map undergoes an intriguing shift towards an unrelated caption in the first two frames (c). Moving on to the middle-left example in Figure 2.16, the model initially demonstrates a keen understanding of the xylophone's location where the sound originates (b). However, subsequent training on additional tasks weakens auditory attention, and the model fails to locate the sounding region (c). This challenge becomes more pronounced when multiple sounding objects are involved. In the middle-right example in Figure 2.16, we explore a scenario where a child is singing alongside a man playing the guitar. The initial visual attention map correctly identifies both the guitar and the child's mouth. Nevertheless, as the model undergoes continuous training, the correlation between the singing voice and the child's visual presence diminishes, and the model connects the sound with the guitar only (c). Similarly, in the bottom-left example of Figure 2.16, the visual attention map shifts from the horse to the human, accompanied by the weakening of auditory attention towards the horse's clip-clop sound (b). Lastly, in the bottom-right example of Figure 2.16, despite the presence of only one prominent sounding object, the bird, the visual attention map is activated at the uncorrelated object. However, our approach successfully mitigates this forgetting problem, as demonstrated in (d) of the example, where the attention maps remain consistent with the initial attention maps.

Figure 2.16: **Visualization of cross-attention maps.** **(a)** Examples of raw data pairs. We visualize cross-attention maps of the pairs in **(b)**. The closer the color is to red, the higher the attention score. While the baseline model using *DER++* attends to entirely different parts as can be seen in **(c)**, our method attends to a similar part even after being trained on two additional tasks as presented in **(d)**. The wrong attention region is marked in an orange circle.

This Chapter is based on the work that is submitted to EMNLP 2024.

Chapter 3. Concept-skill Transferability-based Data Selection for Large Vision-Language Models

Instruction tuning, or supervised finetuning on extensive task-specific data, is necessary for Large Vision-Language Models (LVLMs) to generalize well across a broad range of vision-language (VL) tasks. However, training on large VL datasets can become prohibitively expensive. In this work, we introduce COINCIDE, an effective and scalable data selection technique that uses a small model as a reference model to select visual instruction tuning data for efficient finetuning of a target LVLM, focusing on diversity and transferability. Specifically, we cluster the training data using internal activations from a small model, which identifies VL concept-skill compositions needed by a target LVLM. We then sample data from these diverse clusters by considering their density and transferability, or the ability to transfer well to other concept-skill compositions. This approach ensures the diversity of these compositions, which is vital for LVLM generalization. Extensive experiments demonstrate that COINCIDE achieves superior performance and data selection efficiency against 8 strong baselines on two distinct datasets: LLaVA-1.5 and Vision-Flan. Using only 20% of the LLaVA-1.5 dataset, COINCIDE achieves performance comparable to the LVLM finetuned on the whole dataset, with 70% reduction of the wall-clock running time. On the Vision-Flan dataset, our method achieves superior results with only 16.7% of the dataset.

3.1 Introduction

Large Vision-Language Models (LVLMs) [56, 57, 58, 59] are often built by (1) pretraining on paired image-caption datasets and (2) subsequent finetuning on image-instruction data on diverse vision-language (VL) tasks. The second step, referred to as visual instruction tuning (VIT), substantially enhances multimodal instruction-following capabilities. To achieve broad generalization, recent works [60, 61, 62, 63] integrate an increasing number of VL tasks into VIT.

However, training on extensive VIT data incurs significant computational cost, making the process infeasible for small academic labs and individual researchers. Additionally, it is not clear if all the VIT data are necessary for good generalization, as different VL tasks have different abilities to transfer to downstream tasks [64, 65, 66].

In this paper, we investigate the selection of a coreset, a subset that approximates the performance of the full dataset, from large VIT datasets. Conventional coreset selection approaches [67, 68, 69] usually measure training data quality with a score metric to select valuable training data. However, we discover a mismatch between these score metrics and the highly diverse nature of VIT datasets. Due to the divergence within VIT datasets, selecting any subset based on a single metric leads to a coreset dominated by a few tasks. As shown in Figure 3.1, the coreset from the middle of EL2N [5] score distribution consists of samples mostly from only 3-4 tasks. This bias severely reduces the diversity of the selected coreset and, in our experiments, weakens LVLM generalization (Table 3.1).

Instead of coreset selection at the dataset level, which involves applying a single score metric across the entire dataset, we propose to select data at the level of data clusters, which roughly corresponds to compositions of VL concepts and skills. For example, a concept could be street signs or trains on a

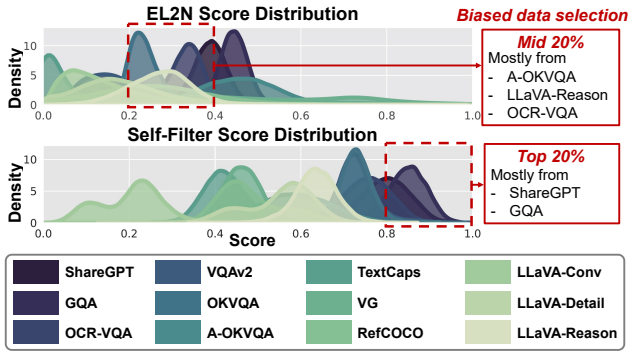


Figure 3.1: **Biased coreset selection.** Different VL tasks in LLaVA-1.5 [4] exhibit different score distributions. Thus, selecting data based on a single score metric like EL2N [5] or Self-Filter [6] results in a biased coreset (red), substantially decreasing the diversity within the coreset.

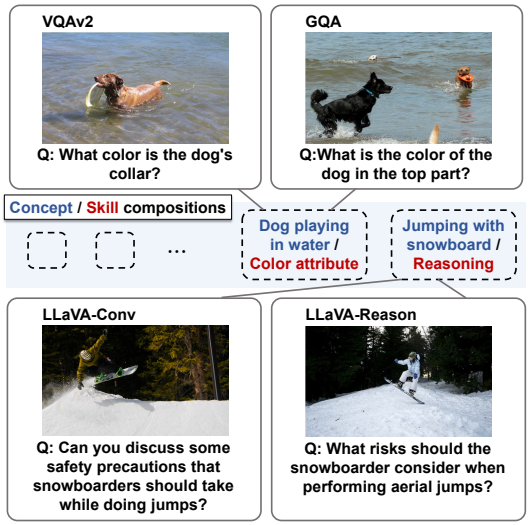


Figure 3.2: **Shared VL concept-skill compositions.** VL tasks (e.g., VQAv2 and GQA) share VL concept-skill compositions.

railroad, while a skill could be OCR, recognizing color, or reasoning. Upon close inspection, we find that different VL tasks contain overlap over these concept-skill compositions. As exemplified in Figure 3.2, LLaVA-Conv and LLaVA-Reason contain questions about the risks of snowboard jumps, despite their separate focuses on multi-turn conversations and reasoning. This suggests sampling over the clusters would be more effective in enhancing the diversity of VL concept-skill compositions than sampling over datasets or tasks.

To this end, we introduce **CO**re **IN**struction **C**oncept-sk**I**ll **D**ata **E**lection (COINCIDE), which identifies VL concept-skill compositions through data clustering using activations from an off-the-shelf, small VLM (Figure 3.3 Left). From each cluster, COINCIDE selects training data for a target LVLM by considering transferability (i.e., how well knowledge from each cluster can facilitate LVLM's learning in other clusters) and internal density of clusters (Figure 3.3 Right). Empirically, we find that transferability correlates well with cosine similarity among clusters. Based on the findings, we select more data points from more transferable clusters. Further, we sample fewer data points from denser clusters, as data points in dense clusters are likely redundant. By selecting data from diverse clusters, COINCIDE enhances the diversity of VL concept-skill compositions in the selected data, leading to better LVLM generalization.

Another major challenge of coreset selection is its high computational cost. Existing techniques often require expensive steps like additional training [70, 71, 6], gradient calculation [72, 73], or the use of bigger and more advanced models [69, 74]. The time complexity and the assumption of larger models contradict the primary goal of coreset selection, which is to reduce the development cost of new models larger than existing ones. In comparison, COINCIDE assumes only a VLM (2B) smaller than the target LVLM (7B, 13B) and does not require any backward pass.

We validate the effectiveness of COINCIDE across a wide range of coreset selection scenarios using two distinct VIT datasets, LLaVA-1.5 [4] and Vision-Flan [7]. The experimental results demonstrate that our method achieves performance competitive with that of the LVLM finetuned with the full dataset, with 30% of time cost including the data selection and training. Our approach also achieves superior performance and efficiency compared to 8 strong baselines.

In summary, our contributions are as follows:

- We introduce COINCIDE, an efficient coreset selection pipeline for a target LVLM using an existing

small reference model to cluster training data. Training on 16.7-20% data selected by COINCIDE achieves comparable performance to whole-dataset finetuning, leading to 70% wall-clock time reduction.

- We propose an efficient transferability calculation among clusters based on our novel observation of a positive correlation between cluster centroid similarity and cluster transferability.
- To enhance training efficacy, we prioritize samples from clusters with high transferability and low density, while still selecting a few samples from other clusters for diversity.

3.2 Related Work

Coreset Selection Coreset selection attempts to extract a subset of training data that functions comparably to the full training set. This technique is adopted for problems like active learning [75, 76], continual learning [77, 78], and data pruning [79, 5]. Recent works [68, 72] investigate coreset selection for instruction tuning of LLMs. Alpagasus [69] uses ChatGPT [80] to rate the quality of instruction samples. S2L [81] leverages the training loss trajectory of smaller models to find optimal samples for training larger LLMs. DiverseEvol [82] utilizes the target model itself to iteratively choose beneficial data for the current training episode.

Coreset Selection for Visual Instruction Tuning Several very recent papers address the coreset selection problem for visual instruction tuning [83, 6, 73]. Self-Filter [6] scores VIT data using a score-net trained along with the target LVLM. The concurrent work TIVE [73] employs gradient information from the target LVLM to compute task- and sample-level importance. Although effective, it demands considerable memory to store the high-dimensional gradient vectors. Moreover, these methods require backward passes, which are expensive due to the large training set. Both also overlook the diversity of selected data, which is vital for generalization. In contrast, our approach reduces wall-clock running time and considers both transferability and diversity.

VL Concept and Skill Discovery Discovering concepts learned by neural networks is a popular topic in interpretability research [84, 85, 86]. Notably, Kowal et al. [87] performs hierarchical clustering in layer-wise activation space. Tiong et al. [64] attempt to identify latent skills underlying VL datasets. Michaud et al. [88] performs spectral clustering to discover the skills of LLMs. Though these works provide inspiration, they are orthogonal to our work, whose main objective is to sample from data clusters rather than understanding existing neural networks. The only application of concept discovery we are aware of is by Gupta et al. [89], who shows enforcing consistent VL concepts improves transfer learning.

3.3 Method

We start by introducing the framework that utilizes neuron activations from a small VLM to group VIT data into clusters, where each cluster comprises samples exhibiting a similar concept-skill composition (Section 3.3.2). Next, we conduct experiments to examine the correlation between the similarity of a cluster centroid to other centroids and the transferability of that cluster to others (Section 3.3.3). Based on our findings, we describe our data selection strategy, which performs cluster-wise sample selection

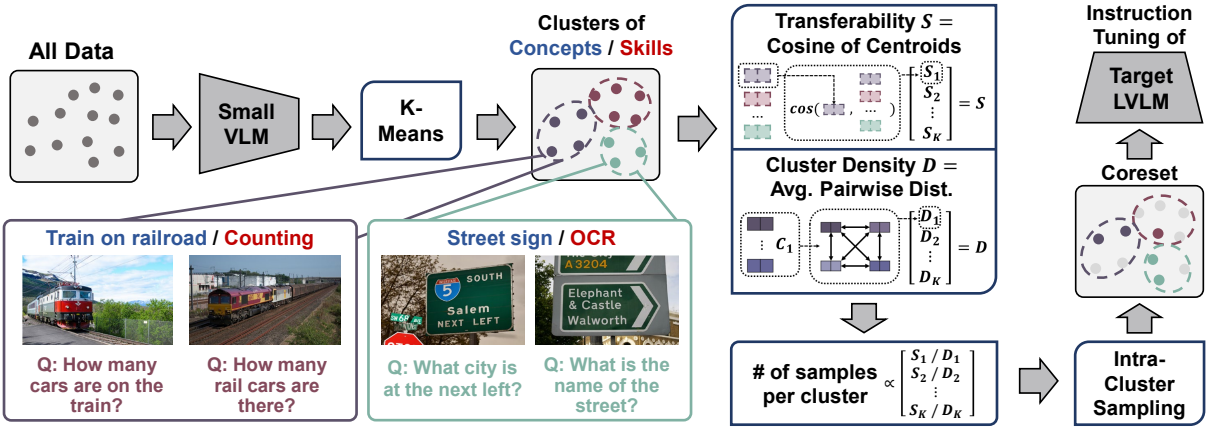


Figure 3.3: **Illustration of COINCIDE.** Our method utilizes a small VLM to cluster visual instruction tuning data based on concept-skill compositions. We then assess the cluster transferability as the mean cosine similarity to other cluster centroids. We further compute the cluster density as the mean Gaussian kernel distance among all data pairs within the cluster. Using cluster transferability and density, COINCIDE determines the number of data to sample from each cluster and performs intra-cluster sampling. Finally, it combines all the selected samples from all the clusters to compose the final coreset.

by selecting different numbers of samples from clusters depending on their transferability and diversity (Section 3.3.4). The overall framework of our approach is illustrated in Figure 3.3.

3.3.1 Preliminaries

A modern LVLM typically consists of a visual encoder and an LLM, which are connected by intermediate network layers. The visual information is fed to the LLM as input [57, 90], or guides cross-attention [91]. We focus on a transformer-based LLM that receives visual information as input tokens.

The l-th transformer layer receives the visual tokens $\boldsymbol{x}_l^v \in \mathbb{R}^{N_v \times D}$ and text tokens $\boldsymbol{x}_l^t \in \mathbb{R}^{N_t \times D}$, where N_v and N_t are the numbers of tokens, and D is the hidden dimension size. A transformer layer contains a multi-head self-attention (MSA) and a feed-forward network (FFN). For the purpose of this paper, we describe only MSA formally:

$$[\boldsymbol{z}_l^v, \boldsymbol{z}_l^t] = \texttt{MSA}_l\left(\texttt{LN}_l\left([\boldsymbol{x}_l^v, \boldsymbol{x}_l^t]\right)\right) + [\boldsymbol{x}_l^v, \boldsymbol{x}_l^t], \tag{3.1}$$

where $[\cdot,\cdot]$ denotes concatenation, $\texttt{LN}_l$ denotes layer normalization, and $\boldsymbol{z}_l^v$ and $\boldsymbol{z}_l^t$ are output visual and text features from the l-th layer MSA, respectively.

3.3.2 Discovering Concept-Skill Compositions

An LVLM aims to learn about a large variety of visual-linguistic concepts and skills. Hence, it is important to automatically sort training data into concepts and skills, so that the coreset can provide sufficient coverage of these. Recent studies [92, 93, 94] reveal that the internal activations at various layers of LVLMs may encode different visual concepts.

To figure out which layer of the LVLM provides the best feature representation for visual concept and skill discovery, we perform a preliminary visualization study of TinyLLaVA-2B [95]. Given an image and a textual question, we visualize the image patches that contribute the most to the generation of the ground-truth answer. Using features from different layers highlights different image patches. Ideally, we can compare the visualization with human intuition and select the layer that agrees with human intuition the most. We provide detailed experimental procedures with some visualization results in Section 3.7.

Perhaps surprisingly, we find that the best layer varies substantially according to the input. That is, the VL concepts and skills are distributed across different layers. Hence, for the clustering, we choose five layers spanning from the initial to top layers of the model to cover a wide range of concepts and skills and use the concatenation of their output as the feature vector of each data point.

We cluster VIT training data points using their feature vector from multiple layers of a small VLM, called a reference model. We extract the features right after the MSA of the l-th layer (Eq. 3.1) and process them into unit-length vectors:

$$\begin{aligned} \boldsymbol{u}_l^v &= \texttt{L2-Normalize}(\texttt{MeanPool}(\texttt{tanh}(\boldsymbol{z}_l^v))), \\ \boldsymbol{u}_l^t &= \texttt{L2-Normalize}(\texttt{MeanPool}(\texttt{tanh}(\boldsymbol{z}_l^t))), \end{aligned} \tag{3.2}$$

where the mean-pooling is performed across the number of visual and text tokens, respectively. The hyperbolic tangent function, `tanh`, is necessary to reduce the impact of a few extreme activations, which are described by Sun et al. [96]. Without this step, these large values would dominate the feature vector and skew the clustering. After that, we concatenate features from the small VLM's layers:

$$\boldsymbol{u}^m = [\boldsymbol{u}_{l_1}^v,\ \boldsymbol{u}_{l_1}^t,\ \ldots,\ \boldsymbol{u}_{l_M}^v,\ \boldsymbol{u}_{l_M}^t] \ / \ \sqrt{2M}, \tag{3.3}$$

where M denotes the number of layers where we extract the features, and the subscripts $l_1, \ldots l_M$ are the layer indices. The resultant $\boldsymbol{u}^m \in \mathbb{R}^{2M*D}$ is the final multimodal feature of the data point.

Then, we perform spherical k-means clustering on $\boldsymbol{u}^m$, yielding K clusters. To ensure the purity of clusters, we set K to a large number, such as $10,000$. Despite its simplicity, the k-means procedure runs in $O(NK)$ time for N data points, which is advantageous when both N and K are large. Other clustering techniques such as spectral clustering or affinity propagation are much more expensive. Qualitative analysis indicates the clusters effectively capture concept-skill compositions. We provide visualization of the clusters in Section 3.8.

3.3.3 Measuring Cluster Transferability

Empirical evidence shows that datasets differ in their ability to generalize to other datasets [97, 98]. We hypothesize that (1) data clusters also have varying levels of transferability and (2) clusters close together in feature space transfer well to each other. If (1) is true, it would be beneficial to select data from highly transferable clusters. If (2) is true, we can use distance among clusters as a proxy for transferability.

We design an experiment to verify the hypotheses. Following Chen et al. [99], to measure transferability from cluster C_i to cluster C_j, we run two training sessions. First, we finetune an LVLM on the same number of samples, N_c, drawn from C_i and C_j respectively. Second, we finetune on N_c samples from C_j only. After finetuning, both models are tested on unseen samples from C_j, yielding test losses $L_{i,j\to j}$ and $L_{j\to j}$. The difference $L_{j\to j} - L_{i,j\to j}$ can be seen as the degree by which C_i facilitates the learning of C_j. We aggregate over target clusters to compute the transferability of the source cluster C_i:

$$T_i = \frac{1}{K_{\text{tgt}}} \sum_{j=1}^{K_{\text{tgt}}} (L_{j\to j} - L_{i,j\to j}), \tag{3.4}$$

where K_{tgt} is the number of target clusters. Then, we compute the cosine similarity of the source cluster

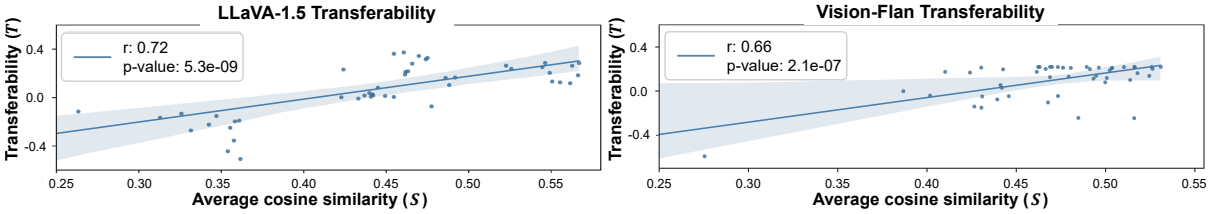


Figure 3.4: **Correlation between cluster centroid similarity and transferability.** We examine the correlations in the LLaVA 1.5 [4] and Vision-Flan [7] datasets, with each point representing a source cluster. We report the Pearson correlation coefficient (r) and p-value.

with the target clusters and average:

$$S_i = \frac{1}{K_{\text{tgt}}} \sum_{j=1}^{K_{\text{tgt}}} \cos(\boldsymbol{e}_i, \boldsymbol{e}_j), \tag{3.5}$$

where $\boldsymbol{e}_i$ is the cluster centroid of cluster C_i.

We compute the correlation between transferability T_i and average cosine similarity S_i over all possible pairings between 50 random source clusters and 50 random target clusters, and plot the results in Figure 3.4. We find that (1) clusters differ significantly in transfer power, and (2) S_i and T_i have a strong positive correlation (0.66-0.72), indicating that the cosine similarity among clusters can serve as an effective and inexpensive proxy for transferability. For K clusters, the time complexity of all cosine similarities is $O(K^2)$. Further studies of transferability are available in Section 3.9.

3.3.4 Data Selection Criteria

In addition to transferability T_i and its proxy S_i, we consider the density of a cluster during the sampling process, as selecting too many data points from a dense cluster that contains many similar samples would create redundancy. Hence, we introduce a density measure D_i:

$$D_i = \frac{1}{|C_i|(|C_i|-1)} \sum_{p,q \in C_i, p \neq q} d(p,q), \tag{3.6}$$

where p and q are two distinct data points from cluster C_i, and $d(p,q) = \exp(-\|\boldsymbol{u}_p^m - \boldsymbol{u}_q^m\|^2)$ is the Gaussian kernel function with $\boldsymbol{u}_p^m$ and $\boldsymbol{u}_q^m$ being the multimodal neuron activations (Eq. 3.3) of data points p and q, respectively. The small D_i value indicates that the cluster C_i is highly diverse.

In order to create a coreset of N_{core} samples, we select from cluster C_i exactly $N_{\text{core}} P_i$ samples. Here, $P_i \propto \exp(S_i/(\tau D_i))$ is a categorical distribution and τ is a temperature hyperparameter. This approach enables us to select more samples from more transferable and less dense clusters to enhance training efficacy, while still selecting a few samples from other clusters to ensure diverse concept-skill compositions in the coreset.

From cluster C_i, we aim to select $N_{\text{core}} P_i$ samples that are representative of the original data distribution of C_i. We compute the distance between the original cluster C_i and the set of sampled data points C_i' as MMD^2, the squared maximum mean discrepancy, which is defined as:

$$\text{MMD}^2 = A(C_i, C_i) + A(C_i', C_i') - 2A(C_i, C_i'), \quad A(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d(p,q). \tag{3.7}$$

We iteratively add samples from the cluster C_i to the sampled cluster C_i' that minimizes MMD^2 using

greedy search [100]. In the end, we combine all the selected samples from all the clusters to compose the final VIT coreset. The complete data selection algorithm is shown in Section 3.12.

3.4 Experiments

3.4.1 Setup

Visual Instruction Tuning Datasets We conduct coreset selection on two distinct VIT datasets: LLaVA-1.5 [4] and Vision-Flan [7]. The LLaVA-1.5 dataset contains 665k VIT data from 12 different VL tasks. The Vision-Flan dataset comprises 191 VL tasks, each with approximately 1k expert-annotated VIT data points, totaling 186k samples.

Models for Training and Data Selection For the target LVLMs, we use the pre-trained LLaVA-1.5 model [4] with a default size of 7B parameters unless otherwise specified. In all experiments, we train the models using LoRA [101] for one epoch, following the official finetuning hyperparameters specified in LLaVA-1.5. As a reference model, we use the TinyLLaVA-2B [95], a small VLM finetuned on the target VIT dataset, for efficient coreset selection for all methods unless otherwise specified. All experiments are conducted using 4 V100 GPUs.

Evaluation Benchmark To assess the generalization of finetuned LVLMs across diverse visual instructions, we evaluate the models on several widely adopted zero-shot multimodal evaluation benchmarks, including 1) visual question answering: VQAv2 [102], GQA [103], VizWiz [104]; 2) knowledge-grounded QA: ScienceQA [105]; 3) Optical Character Recognition (OCR): TextVQA [106]; 4) hallucination: POPE [107]; 5) multiple-choice: MME [108], MMBench [109]; 6) free-form generation: LLaVA-Bench [90], MM-Vet [110]. In all experiments, we follow the protocols outlined in LLaVA-1.5 and Vision-Flan to select evaluation benchmarks. Further explanations of these benchmarks are provided in Section 3.6.

Since each evaluation benchmark has a different scale, we compute average relative performance, denoted as Rel., across benchmarks to assess the level of generalization. Each relative performance is derived from the formula: (model performance / full-finetuned performance) $\times$ 100%.

Baselines We compare our method with several coreset selection techniques: CLIP-Score, EL2N [5], Perplexity [67], SemDeDup [111], D2-Pruning [112], Self-Sup [113]. We also compare with a recent VIT coreset selection method, Self-Filter [6]. We additionally report the results of *Random*, the model finetuned with the coreset collected by random sampling, and *Full-Finetune*, the model finetuned with the full VIT dataset. The details of the baseline methods are provided in Section 3.6.

3.4.2 Results and Discussion

COINCIDE surpasses baselines on LLaVA-1.5. Table 3.1 presents model performance when we limit the coreset to 20% of the size of the LLaVA-1.5 VIT dataset. COINCIDE is either the best or a close second on 7 out of 10 benchmarks, including VQAv2, GQA, SQA-I, TextVQA, POPE, MME, and MMBench-en. On average, COINCIDE outperforms the best baseline by 1.6 percent points (pp) in relative performance.

Table 3.1: **Comparison of coreset selection techniques on the LLaVA-1.5 dataset.** We finetune the models using coresets with a 20% sampling ratio and estimate performance on various multimodal evaluation benchmarks. The best and the second best results are in **bold** and underlined, respectively.

Method	VQAv2	GQA	VizWiz	SQA-I	TextVQA	POPE	MME	MMBench en	MMBench cn	LLaVA-Bench	Rel.(%)
Full-Finetune	79.1	63.0	47.8	68.4	58.2	86.4	1476.9	66.1	58.9	67.9	100
Random	75.7	58.9	44.3	68.5	55.3	84.7	1483.0	62.2	54.8	65.0	95.8
CLIP-Score	73.4	51.4	43.0	65.0	54.7	85.3	1331.6	55.2	52.0	66.2	91.2
EL2N	76.2	58.7	43.7	65.5	53.0	84.3	1439.5	53.2	47.4	64.9	92.0
Perplexity	75.8	57.0	47.8	65.1	52.8	82.6	1341.4	52.0	45.8	68.3	91.6
SemDeDup	74.2	54.5	46.9	65.8	55.5	84.7	1376.9	52.2	48.5	**70.0**	92.6
D2-Pruning	73.0	58.4	41.9	**69.3**	51.8	85.7	1391.2	**65.7**	**57.6**	63.9	94.8
Self-Sup	74.9	59.5	46.0	67.8	49.3	83.5	1335.9	61.4	53.8	63.3	93.4
Self-Filter	73.7	58.3	**53.2**	61.4	52.9	83.8	1306.2	48.8	45.3	64.9	90.9
COINCIDE (Ours)	**76.5**	**59.8**	46.8	69.2	**55.6**	**86.1**	**1495.6**	63.1	54.5	67.3	**97.4**

Interestingly, all baselines perform worse than the random sampling on average relative performance, suggesting that they may be susceptible to the selection bias, which is discussed in the introduction and illustrated in Section 3.1. In contrast, COINCIDE considers the diversity of VL concept-skill compositions, demonstrating high generalization across a broad range of visual instructions. We further analyze the selection bias of the baselines and effectiveness of COINCIDE in Section 3.10.

In Figure 3.5, we show the performance comparison across different coreset sizes as proportions of the original LLaVA-1.5 dataset. COINCIDE consistently outperforms other baselines across various sampling ratios, underscoring the effectiveness of our approach. COINCIDE also performs well on LLaVA-1.5-13B, as shown in Section 3.11.1.

One Sixth of Vision-Flan selected by COINCIDE outperforms full dataset. We further evaluate the coreset selection techniques on the Vision-Flan VIT dataset [7] and show the results in Table 3.2. COINCIDE exceeds the performance of the model finetuned on the whole Vision-Flan data by 1.0 pp and the performance of the best baseline by 4.5 pp, using a selected subset 16.7% (1/6) of its size. Further, as illustrated in Figure 3.6, COINCIDE maintains consistently high performance across several sampling rates.

We note that Vision-Flan, with its 191 VL tasks, is much more diverse than the LLaVA-1.5 dataset of 12 tasks. The stronger performance of COINCIDE on the Vision-Flan suggests that COINCIDE algorithm is well adapted to the use case of visual instruction tuning, which is increasingly performed on larger and more diverse sets of tasks.

Another curious phenomenon is that several baselines, including CLIP-Score, Perplexity, and Self-Filter, experience performance declines as the sampling ratio increases in Figure 3.6. A similar trend is observed in the random baseline in Figure 3.5. This underscores the importance of coreset selection, as merely increasing the dataset size does not guarantee improved LVLM capabilities.

COINCIDE provides wall-clock training time reduction and is Pareto superior. In Figure 3.7, we plot the wall-clock time cost of the entire pipeline of data selection and model finetuning versus the average relative performance (Rel.) on the LLaVA-1.5 dataset. COINCIDE achieves 97.4%, 98.4%, and 99.4% Rel. with the wall-clock times of 15.1, 25.1, and 35.1 hours, respectively. In contrast, finetuning on all data takes 50 hours.

We observe that COINCIDE provides Pareto superior solutions to all baselines. This is mainly due to the excellent time complexity of COINCIDE, which is linear to the number of training data points. Moreover, our method discovers the transferability among clusters at a low computational cost.

Figure 3.5: **Average relative performances of all techniques at different coreset sizes for the LLaVA-1.5 dataset.**

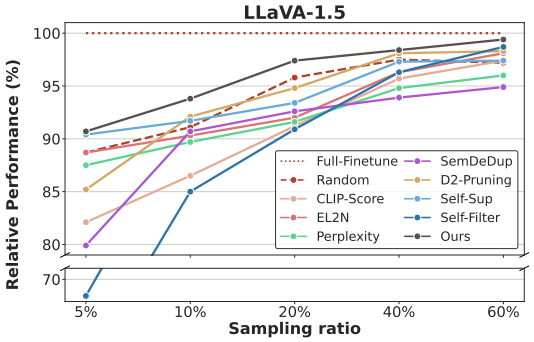


Figure 3.6: **Average relative performances of all techniques at different coreset sizes for the Vision-Flan dataset.**

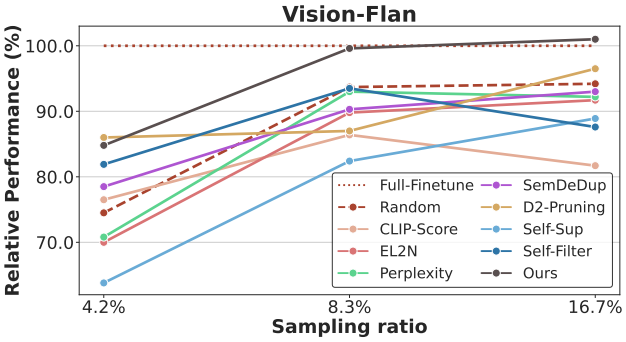


Figure 3.7: **Comparison of coreset selection techniques on average relative performance and wall-clock time cost.** The wall-clock time cost includes both the data selection and finetuning of the target LVLM. The time cost is measured in hours of running time on a computing node with 4× V100 GPUs.

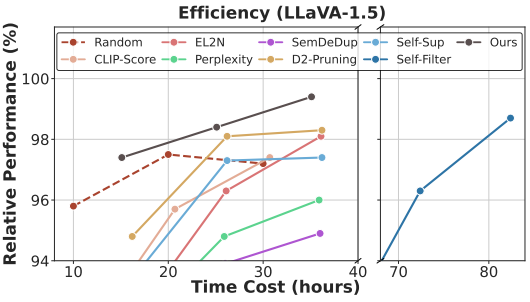


Table 3.2: **Comparison of coreset selection techniques on the Vision-Flan dataset.** We finetune the models using coresets with a 16.7% sampling ratio and estimate performance on various multimodal evaluation benchmarks. The best and the second best results are in **bold** and underlined, respectively.

Method	MMBench-en	MME	MM-Vet	POPE	SQA-I	Rel.(%)
Full-Finetune	53.4	1287.5	25.6	84.2	61.3	100
Random	45.2	1122.3	26.1	82.5	60.9	94.2
CLIP-Score	34.3	687.6	26.6	72.6	61.8	81.7
EL2N	45.3	1082.9	23.9	82.1	60.6	91.7
Perplexity	39.3	1160.9	26.1	83.1	59.2	92.2
SemDeDup	42.1	1146.5	27.2	82.7	56.8	93.0
D2-Pruning	49.1	1052.4	27.0	82.5	**64.7**	96.5
Self-Sup	42.9	1012.2	23.5	80.8	60.0	88.9
Self-Filter	28.6	923.6	**30.0**	**83.3**	59.3	87.6
COINCIDE (Ours)	**56.7**	**1222.2**	26.2	81.9	63.8	**101.0**

It requires only cosine similarity calculations, with a time complexity quadratic to the number of clusters. Hence, COINCIDE provides a scalable data selection procedure.

COINCIDE also utilizes neuron activations from intermediate layers of the small reference model rather than the final outputs, avoiding complete forward passes like other baselines. Additionally, COINCIDE does not require training of additional networks that score data points, like Self-Filter. Neither does it require backward passes like the concurrent work TIVE [73]. The combination of all these factors leads to an efficient solution to coreset selection.

3.4.3 Further Analysis and Ablation

Alternative Reference Models We analyze the effects of different reference models, which are the models used to extract features for clustering and cosine similarity. We compare four models, CLIP, TinyLLaVA-0.9B, TinyLLaVA-2B, and LLaVA-1.5-7B, and report the time cost of the entire coreset selection pipeline and average relative performance in Table 3.3 (a). We observe that CLIP performs the worst whereas TinyLLaVA-2B performs the best with reasonable time cost in data selection. However, the differences between TinyLLaVA-0.9B, TinyLLaVA-2B, and LLaVA-1.5-7B are small. We conclude that a well-trained small model can serve effectively as a reference model in coreset selection for a target LVLM. We also examine the robustness of COINCIDE when the reference model is finetuned on a different VIT dataset, which is detailed in Section 3.11.2.

Table 3.3: **Ablation studies of COINCIDE. (a)** Effect of different reference models. The time cost includes both the data selection and finetuning of the target LVLM and is measured in hours of running time on a computing node with 4× V100 GPUs. **(b)** Ablation on data selection criteria of our approach, transferability (S) and density (D). **(c)** Performances of different intra-cluster sampling strategies across various coreset sizes.

(a) Reference Model

Model (# params)	Time (hours)	Rel. (%)
CLIP (0.4B)	**10.9**	94.2
TinyLLaVA (0.9B)	12.2	96.3
TinyLLaVA (2B)	15.3	**97.4**
LLaVA-1.5 (7B)	20.7	97.1

(b) Key Components

Method	S	D	Rel.(%)
Random	–	–	95.8
COINCIDE (Ours)	–	–	94.4
	✓	–	95.9
	–	✓	94.7
	✓	✓	**97.4**

(c) Intra-Cluster Sampling methods

Intra-Cluster Sampling	Sampling ratio				
	5%	10%	20%	40%	60%
Random-select	90.1	**94.3**	**97.5**	97.7	98.3
Nearest-to-centroid	**91.9**	**94.3**	96.7	**99.1**	98.4
Greedy-MMD^2-minimize	90.7	93.8	**97.4**	98.4	**99.4**

Ablation on Data Selection Criteria To validate our coreset selection method, we conduct ablation studies on the two data selection criteria, transferability and density, as summarized in Table 3.3 (b). In the first ablation, without using either criterion, we simply select the same number of samples from each cluster. This results in inferior performance, which suggests that naive stratified sampling from the clusters is not sufficient, possibly due to the heterogeneous nature of the clusters. In the second ablation, number of samples from each cluster is proportional to the transferability of the cluster, leading to a 1.5 percentage point (pp) increase. The third ablation selects number of samples inversely proportional to density, yielding a modest enhancement of 0.3 pp. Finally, combining both transferability and density provides a sizeable increase of 3.0 pp, demonstrating that the two selection criteria are complementary to each other.

Intra-cluster Selection Criteria COINCIDE selects samples within a cluster by minimizing MMD^2. We examine the effects of two alternative techniques, random selection and selecting samples closest to the centroid. As shown in Table 3.3 (c), in small coresets, samples closest to the centroids, which are probably not outliers or hard samples, lead to high performance. In contrast, under high sampling ratios (i.e., large coresets), selecting diverse data using the MMD^2 metric leads to high performance. This is reminiscent of the finding of Sorscher et al. [113] that easy samples are beneficial when the sampling ratio is small, whereas hard samples are advantageous when the sampling ratio is large. Overall, the COINCIDE algorithm is robust to the choice of intra-cluster sampling.

3.5 Conclusion

In this paper, we introduce COINCIDE, a cluster-level visual instruction tuning data selection for efficient finetuning of Large Vision-Language Models. We demonstrate that clustering based on inner activations from a small model can represent visual-linguistic concept-skill compositions shared among diverse tasks in visual instruction tuning datasets. Additionally, our empirical investigation validates a strong positive correlation between cosine similarity and transferability among clusters. Based on the transferability and density of clusters, COINCIDE selects more samples from more transferable and less dense clusters to enhance training efficacy, while preserving the diversity of concept-skill compositions within the coreset to ensure better model generalization ability. Comprehensive experiments on the LLaVA-1.5 and Vision-Flan datasets demonstrate that our method outperforms strong baselines across several benchmarks with the lowest data selection cost, showcasing both the effectiveness and efficiency of our approach.

Limitations

In our experiments, we observe that VL concept-skill compositions are shared across various VL tasks and identify VL concept-skill compositions that transfer well to others. However, after identifying these compositions and performing coreset selection, we finetune the target LVLMs by randomly selecting samples from the coreset. Recognizing the growing research attention on the importance of training order in LLM instruction tuning, we believe that considering the training order for LVLMs is crucial to enhance efficiency in visual instruction tuning. In future research, we aim to develop a curriculum learning algorithm that automatically determines the optimal training order based on the identified VL concept-skill compositions to further reduce the development cost of a new model.

Additionally, we assess whether the data with similar concept-skill compositions are concentrated well on the clusters through human inspection. Therefore, further investigation should be conducted to quantitatively evaluate the clustering of data with similar concept-skill compositions. Quantitative measures are expected to enable more accurate identification of VL concept-skill compositions and their transferability.

Ethics Statement

In this work, we use publicly available visual instruction tuning datasets for coreset selection to enable easy replication. However, some data in the datasets contain erroneous answers about the visual content or images that do not clearly connect with the provided answers. Finetuning Large Vision-Language Models (LVLMs) with such data conveys wrong interpretations, inducing hallucinations in the LVLMs. Hallucination in LVLMs refers to a phenomenon where the LVLMs generate descriptions that are inconsistent with the target image. This poses a significant ethical issue for deploying LVLM in real-world applications. However, current coreset selection techniques, including ours, do not address hallucination in their selection processes. This motivates further research in coreset selection to identify visual instruction tuning data that minimizes hallucinations, aiming to build more reliable and trustworthy LVLMs.

3.6 Details of Experimental Setups

Evaluation Benchmark We provide in-depth explanations of the multimodal evaluation benchmarks used in our experiments. (1) VQAv2 [102] evaluates the ability to understand and reason about general visual content by answering open-ended questions based on images. (2) GQA [103] assesses compositional reasoning and understanding skills, requiring models to understand relationships and attributes of objects within images. (3) Vizwiz [104] is designed to evaluate the model's ability to cope with real-world visual impairments. (4) ScienceQA-Image (SQA-I) [105] tests the model's science-related reasoning and visual understanding of images. (5) TextVQA [106] specifically targets text in images, assessing the Optical Character Recognition (OCR) ability of models. (6) POPE [107] measures object hallucination in models. (7) MME [108] contains binary choice questions designed to evaluate perception and cognition abilities through 14 subtasks. (8) MMBench [109] evaluates various abilities of models, covering object detection, text recognition, relation reasoning, etc., using tests conducted in English (en) or Chinese (cn). (9) LLaVA-Bench [4] is specifically designed for evaluating models on visual instruction-following and chat ability. (10) MM-Vet [110] measures VL capabilities, including recognition, OCR, knowledge, language

Table 3.4: **Hyperparameter configurations.**

Method	LLaVA-1.5	Vision-Flan
CLIP-Score	top score selected	top score selected
EL2N	medium score selected	medium score selected
Perplexity	medium score selected	medium score selected
SemDeDup	$\boldsymbol{K}$: 10,000	$\boldsymbol{K}$: 5,000
D2-Pruning	$\boldsymbol{k}$: 5, γ_r : 0.4, γ_f : 1.0	$\boldsymbol{k}$: 5, γ_r : 0.4, γ_f : 1.0
Self-Sup	$\boldsymbol{K}$: 10,000	$\boldsymbol{K}$: 5,000
Self-Filter	$\boldsymbol{k}$: 10, γ : 1	$\boldsymbol{k}$: 10, γ : 1
COINCIDE (Ours)	$\boldsymbol{K}$: 10,000, τ : 0.1	$\boldsymbol{K}$: 5,000, τ : 0.1

generation, spatial awareness, and math.

Baselines In this section, we provide a more detailed explanation of the baselines. The hyperparameters for baselines in our experiments are summarized in Table 3.4.

- **CLIP-Score** utilizes the CLIP [58] model to assess the alignment between images and their instructions. For our study, we select VIT data with the highest CLIP scores.
- **EL2N** [5] estimates sample quality using the Error L2-Norm score, defined as $\mathbb{E}[||p(x) - y||_2]$. Here, $p(\cdot)$ represents the reference model, x is the input, and y is the ground-truth label. This metric calculates the average L2 distance between the model's predictions and the ground-truth labels for text tokens.
- **Perplexity** [67] measures the average negative log-likelihood of the next token prediction, defined as $\exp(-\mathbb{E}[\log p(x)])$. This metric assesses the uncertainty in the model's predictions. For both EL2N and Perplexity, we select data from the middle score distribution, as this range has been shown to perform best in prior research [67].
- **SemDeDup** [111] removes semantically duplicated data by clustering the output embeddings of the last token from the reference model's final layer. This helps in reducing redundancy in the selected coreset.
- **D2-Pruning** [112] represents the dataset as a graph where nodes represent sample difficulty and edges represent distances between samples. It actively uses the graph to preserve diversity in the coreset. We use the AUM [79] score to indicate difficulty, defined as $p_y(x) - \max_{i \neq y} p_i(x)$, where $p_y(x)$ is the prediction value for the ground-truth label, and $\max_{i \neq y} p_i(x)$ is the highest prediction value for any non-ground-truth label. For the distances between samples, we calculate the L2 distance between averaged output embeddings from the last layer tokens of the reference model.
- **Self-Sup** [113] clusters the data using the averaged output embeddings from the last layer tokens of the reference model. It scores data based on their distance to cluster centroids, selecting those the most likely to be prototypical.
- **Self-Filter** [6] is a recent VIT coreset selection method that was originally applied to the LLaVA-158k VIT dataset [90], which consists of only three VL tasks. It finetunes the score-net along with the target LVLM on the full dataset to serve as a reference model for scoring and filtering VIT data. We use the version that additionally incorporates both CLIP scores and CLIP features since it ensures enhanced performance and efficiency.

3.7 Visualizing LVLM Skills with Relevancy Maps

In our method, we extract neuron activations from various layers (Eq. 3.2) to represent the concepts and skills of each VIT data. In this approach, we hypothesize that distinct layers represent distinct concepts and skills of the LVLM. To support this assumption, we compute relevancy maps [114] following the approach outlined in Stan et al. [115]. The relevancy maps help us understand the model's final output by highlighting the most contributing parts of the input for each layer. Given the target output token $\boldsymbol{y_t}$ and the attention map $\boldsymbol{A}_l \in \mathbb{R}^{h \times (N_v+N_l) \times (N_v+N_l)}$ of the $\boldsymbol{l}$-th layer, where h is the head dimension of the attention, the relevancy map $\boldsymbol{R}$ is computed as follows:

$$\begin{aligned} \bar{\boldsymbol{A}}_l &= \mathbb{E}_h[\nabla \boldsymbol{A}_l \odot \boldsymbol{A}_l],\ \nabla \boldsymbol{A}_l = \frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{A}_l}, \\ \boldsymbol{R} &= \boldsymbol{R} + \bar{\boldsymbol{A}}_l \cdot \boldsymbol{R}, \quad \text{for } l \in \{1, 2, \ldots, L\}, \end{aligned} \tag{3.8}$$

where $\odot$ denotes the Hadamard product and L is the total number of layers in the LVLM. In order to investigate the contribution of each layer to the final output, we visualize the image regions related to the output token through the visual relevancy map computed from each layer. Specifically, we consider the row of $\bar{\boldsymbol{A}}_l \cdot \boldsymbol{R}$ corresponding to the output token. Then, we extract the visual token parts of the row to yield the visual relevancy map.

For the investigation, we inspect the 4th, 8th, 12th, 16th, and 20th layers of the TinyLLaVA-2B [95] model and identify the layer that activates the most relevant visual regions. The results, shown in Figure 3.10, reveal that (1) the most relevant layer varies according to the concept-skill composition and (2) the most relevant layer is the same across diverse VIT data when the data shares a similar concept-skill composition. These findings support our initial assumption that different layers contribute to distinct concepts and skills. Therefore, using neuron activations from diverse layers can effectively group VIT data according to their concept-skill composition.

3.8 Concept-Skill Clustering Visualization

We visualize the clustering results of the gathered VIT data. The results are illustrated in Figure 3.11. We observe that most clusters contain VIT data that encode similar concept-skill compositions. For instance, the first group in Figure 3.11 consists of samples requiring OCR and counting abilities to solve visual queries involving images with store signs. The second group features images of people waiting for public transportation and multiple-choice questions that require visual recognition and reasoning abilities. The third group shows a cluster of samples with images of people in suits and queries focusing on object localization and generating captions for given bounding boxes. Lastly, the bottom group includes images exhibiting children with animals and requiring the ability to reason about the educational benefits that the children might gain from interacting with the animals.

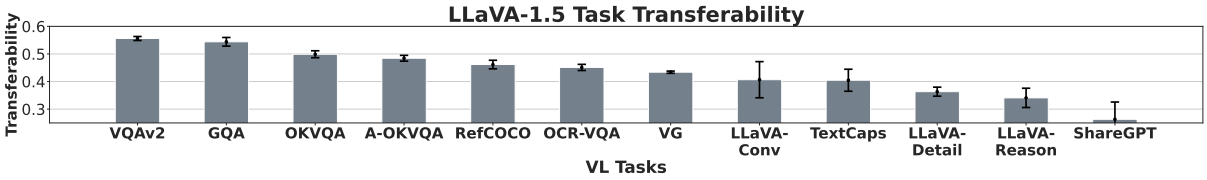


Figure 3.8: **Task-wise transferability.** We group the VIT data by task names and average the cluster transferability of each data.

3.9 In-Depth Analysis on Concept-Skill Composition Transferability

3.9.1 Task-wise Transferability

To further understand transferability, we calculate the transferability of LLaVA-1.5 tasks by averaging the cluster transferability of VIT data. We show the results in Figure 3.8. We observe that VQA tasks, including VQAv2, GQA, OKVQA, and A-OKVQA, contain VIT data that transfers well to other data. In contrast, GPT-generated conversational tasks, including LLaVA-Conv, LLaVA-Detail, LLaVA-Rason, and ShareGPT, exhibit low transferability. This corresponds to the findings of Tiong et al. [64] that VQA tasks are effective for finetuning LVLMs. This alignment supports the efficacy of our approach in discovering the fine-grained concept-skill compositions and their transferability. We hypothesize that the high transferability of the VQA tasks is because these tasks mostly require abilities close to the fine-grained VL concepts and skills that can be shared with other tasks, as described in Figure 3.2, unlike more complex tasks.

3.9.2 Concept-Skill with High Transferability

In Figure 3.12, we visualize concept-skill compositions having the highest transferability for various VL task types. We define the VL task type of a cluster based on the task name associated with most of the cluster's data (e.g., VQAv2, GQA). Interestingly, GQA and LLaVA-Conv share a similar concept-skill composition as their most transferable concept-skill composition. This suggests that the transferability of VL concept-skill composition might be consistent across different VL tasks.

3.9.3 Concept-Skill as Latent Factor of LVLM

We conduct an ablation study to verify if data clusters from different VL task types have high transferability with each other when they share a similar concept-skill composition. In this study, we select two clusters from different VL task types with a similar concept-skill composition (second and fourth groups in Figure 3.12), using the first cluster as the source and the second cluster as the target. Additionally, we employ 49 randomly selected source clusters and measure transferability from the source clusters to the target cluster (Eq. 3.4). The source cluster, sharing a similar concept-skill composition with the target, ranks in the top 5 of the 50 source clusters in terms of test loss gain, exhibiting high transferability to the target cluster. This suggests that concept-skill compositions resemble fine-grained latent factors that constitute LVLM abilities. Thus, these fine-grained VL concepts and skills must be considered to effectively reduce data redundancy and build a well-generalized LVLM.

Table 3.5: **Transfering to the larger target model.** We validate if the coresets selected from TinyLLaVA-2B are transferable to LLaVA-1.5-13B finetining. We train the LLaVA-1.5-13B using coresets with 20% sampling ratio and estimate performance on various multimodal benchmarks. The best and the second best results are highlighted in **bold** and underline, respectively.

Method	VQAv2	GQA	VizWiz	SQA-I	TextVQA	POPE	MME	MMBench en	MMBench cn	LLaVA-Wild	Rel(%)
Full-Finetune	80.0	63.3	58.9	71.2	60.2	86.7	1541.7	68.5	61.5	69.5	100
Random	76.7	**60.5**	48.0	68.8	57.7	84.8	1484.9	62.8	55.2	68.6	94.0
CLIP-Score	75.3	52.6	42.2	69.7	57.3	85.4	1426.3	60.4	54.0	68.1	90.7
EL2N	77.2	59.6	**54.8**	69.9	56.1	84.1	**1531.0**	59.3	52.3	65.8	93.8
Perplexity	77.0	58.5	48.2	68.7	54.8	83.1	1508.8	57.5	50.3	68.7	91.6
SemDeDup	75.6	57.5	48.3	**70.5**	57.7	85.3	1397.6	59.0	51.1	68.7	91.9
D2-Pruning	73.9	**60.5**	49.8	70.4	55.2	84.9	1463.0	**67.3**	**59.9**	66.5	94.7
Self-Sup	76.3	**60.5**	50.0	70.2	52.7	85.4	1463.8	63.7	57.6	64.9	93.6
Self-Filter	75.0	59.8	48.6	69.5	55.8	84.5	1446.9	58.8	51.8	**69.1**	92.2
COINCIDE (Ours)	**77.8**	60.4	51.6	70.0	**58.6**	**87.1**	1516.8	64.0	57.7	67.4	**95.9**

Table 3.6: **Impact of a reference model training dataset.** We use TinyLLaVA-2B finetuned on the LLaVA-1.5 dataset as a reference model to collect coresets from the Vision-Flan dataset with 16.7% sampling ratio. The best and the second best results are highlighted in **bold** and underline, respectively.

Method	MMBench-en	MME	MM-Vet	POPE	SQA-I	Rel.(%)
Full-Finetune	53.4	1287.5	25.6	84.2	61.3	100
EL2N	41.8	1082.0	23.9	82.6	61.7	90.9
Perplexity	45.7	1001.7	26.1	81.9	**64.8**	93.7
SemDeDup	46.8	1129.7	**27.2**	82.5	64.3	96.9
D2-Pruning	48.1	**1143.0**	27.0	83.4	63.1	97.3
Self-Sup	47.1	1084.6	23.5	81.7	63.5	93
COINCIDE (Ours)	**51.7**	1139.0	26.9	**84.0**	64.5	**99.1**

3.10 Concept-Skill Diversity within Coresets

Our method selects data from various clusters to ensure a high diversity of VL concept-skill compositions within the coreset. To demonstrate the efficacy of our method, we compare the diversity within the coreset by our method with those by the baseline methods. Specifically, we use the 191 tasks from the Vision-Flan dataset as proxies for different concept-skill compositions, as there are no ground-truth compositions. We then count the number of selected samples for each task. The results, summarized in Figure 3.13, indicate that baseline methods select most data from only a few tasks, leading to biased selection and undermining LVLM generalization. This bias explains why most baselines perform worse than random sampling in our experiments. In contrast, our method achieves a more balanced selection across the various tasks.

3.11 Additional Experimental Results

3.11.1 Transfering to Larger Target Model

We evaluate the performance of the larger target model (LLaVA-1.5-13B) finetuned on coresets gathered by the small VLM (TinyLLaVA-2B). 3.5 summarizes the performances across various benchmarks. The results demonstrate the effectiveness of our method in selecting a coreset that can be successfully transferred to the larger target model.

Figure 3.9: **Hyperparameter search.** We examine the effect of the temperature (τ) and the number of clusters (K).

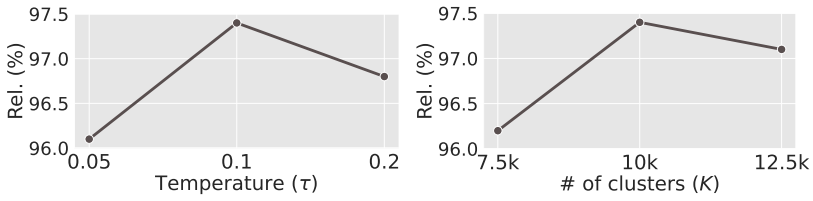


Table 3.7: **Choice of neuron activations.** We investigate the impact of multimodal neuron activations.

Neuron Activation	Rel.(%)
Boolean	95.7
Last layer	96.5
MSA layers	**97.4**
FFN layers	96.0

3.11.2 Robustness of Reference Model

We investigate the robustness of our method when the reference model is finetuned on a VIT dataset different from a target VIT dataset. To this end, we use the TinyLLaVA-2B finetuned on the LLaVA-1.5 VIT dataset, to perform coreset selection from the Vision-Flan dataset. The results are summarized in Table 3.6. COINCIDE continues to show performance comparable to full-finetuning while outperforming other baseline methods.

3.11.3 Hyperparameters

We conduct ablation studies on hyperparameters of our method, which include the number of clusters ($\boldsymbol{K}$) and the temperature ($\boldsymbol{\tau}$). The results, summarized in Figure 3.9, reveal that a sufficiently large number of clusters is essential to ensure cluster purity and diversity of VL concept-skill compositions, ensuring effective representation of the compositions and enhancing the generalization ability of LVLM. Furthermore, we find that setting the temperature too low leads to a biased coreset selection, as most samples are then selected from a few clusters. This undermines the diversity within the coreset, leading to a decline in overall performance.

3.11.4 Multimodal Neuron Activation

We further analyze the impact of different multimodal neuron activations on the performance of our method. COINCIDE selects neuron activations from the MSA blocks across the 4th, 8th, 12th, 16th, and 20th layers of the reference model. We experiment with different neuron activations and present the results in Table 3.7. Transforming the neuron activations from the MSA blocks into boolean vectors by mapping negative values to -1 and positive values to 1 causes a significant performance drop, likely due to substantial information loss, yielding inaccurate clustering and transferability calculation. Extracting neuron activations only from the last layer of the reference model causes a slight performance decrease. As discussed in Section 3.3.2, LVLM abilities stem from various layers. Hence, relying on the last layer captures only a small portion of these capabilities, leading to the performance decline. Finally, utilizing the neuron activations from the MSA blocks gives superior performance compared to using activations from the FFN blocks. We believe this is because MSA layers use self-attention to share multimodal information, providing richer multimodal understanding.

3.12 The COINCIDE Algorithm

In Algorithm 4, we outline our VIT data selection procedure, which involves several key stages: clustering the data (lines 1-2), calculating the cluster categorical distribution (lines 3-5), and selecting samples from each cluster (lines 6-15).

Algorithm 4 COINCIDE Data Selection Algorithm

Require: K: the number of clusters, N_{core}: target coreset size

Extract multimodal neuron activations $\boldsymbol{u}^m$ from the full dataset. ▷ Eq. 3.3
Cluster $\boldsymbol{u}^m$ into K clusters to form a set of clusters $\mathcal{C}=\{\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_K\}$.
Compute cluster transferability $S_i=\mathbb{E}_j\left(\texttt{cos}(\boldsymbol{e}_i,\boldsymbol{e}_j)\right),\ i\in\{1,2,\ldots,K\}$ ▷ Eq. 3.5
Compute cluster density $D_i=\mathbb{E}_{p,q\sim C_i}\left(d(p,q)\right),\ i\in\{1,2,\ldots,K\}$ ▷ Eq. 3.6
Calculate cluster categorical distribution $P_i \propto \exp(S_i/(\tau D_i))$.
for $i=1,2,\ldots,K$ **do**
    i-th cluster empty coreset $\mathcal{C}'_i$.
    i-th cluster target sample size $N_{\text{core},i}=N_{\text{core}}P_i$.
    while $|\mathcal{C}'_i| < N_{\text{core},i}$ **do**
        $k=\underset{j\in\mathcal{C}_i\setminus\mathcal{C}'_i}{\text{argmin}}\ \text{MMD}^2\left(\mathcal{C}_i,\mathcal{C}'_i\cup\{j\}\right)$ ▷ Eq. 3.7
        $\mathcal{C}'_i \leftarrow \mathcal{C}'_i\cup\{k\}$
    end while
end for
return $\mathcal{C}'_1\cup\mathcal{C}'_2\cup\ldots\cup\mathcal{C}'_K$

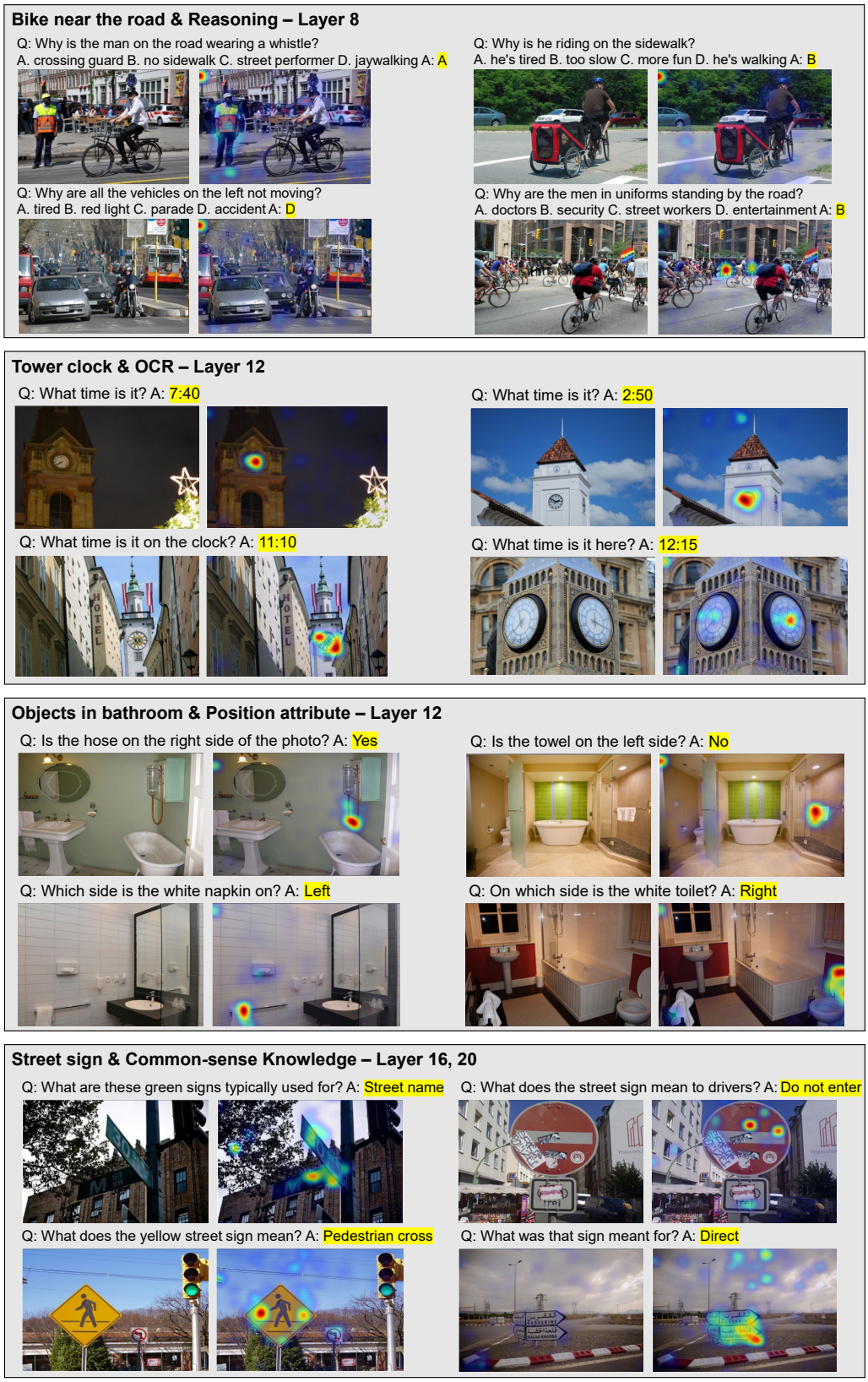


Figure 3.10: **Relevancy maps visualization.** We investigate which layer contributes most to the final output of the LVLM. This is done by visualizing relevancy maps of four samples from the same cluster. For each example, the left image is the original, while the right image shows the visualized relevancy map, highlighting regions most relevant to the LVLM output text colored in yellow. The top-left corner of each group explains the VL concept-skill composition and the layer number with the highest relevancy to the output.

Figure 3.11: **Examples of data clusters.** We visualize four samples from the same cluster. The top-left corner of each group explains the VL concept-skill composition.

Figure 3.12: **High transferability cluster sample visualization.** We visualize the samples from the most transferable concept-skill composition for each VL task. The top-left corner of each group explains the VL task type and the VL concept-skill compositions. The VL task type for the group follows the task name where most of the data from the group are associated.

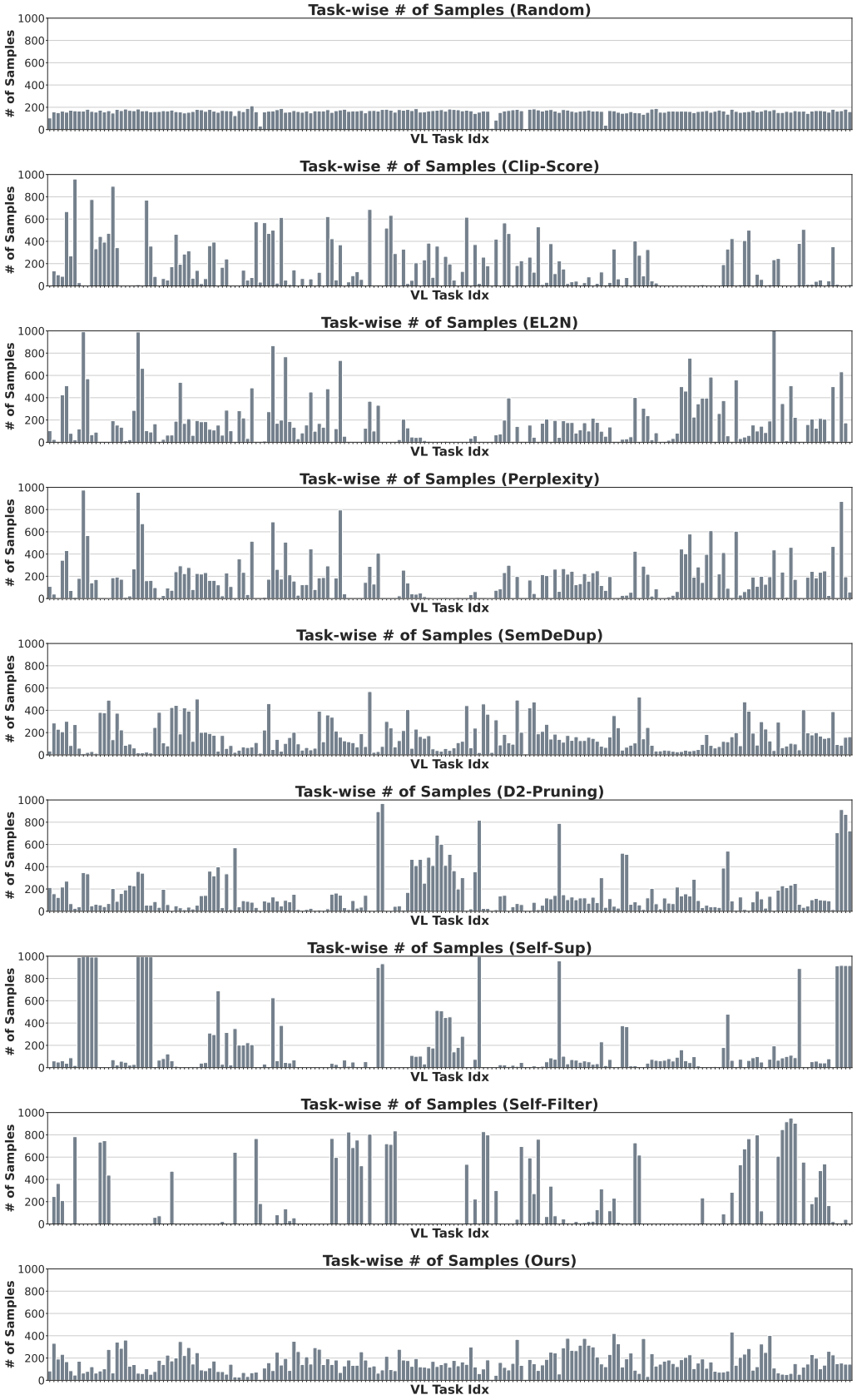


Figure 3.13: **Task-wise numbers of samples.** The number of selected samples per VL task in the Vision-Flan VIT dataset. The horizontal axis denotes the VL task index in the dataset, and the vertical axis denotes the number of samples. Baseline methods result in biased coresets. In contrast, our method achieves a more balanced sample selection across diverse tasks, leading to better LVLM generalization.

Chapter 4. Concluding Remark

In this thesis, we focus on efficient techniques for multimodal learning. Our goal is to address the substantial computational costs and memory requirements involved in retraining models from scratch to keep models up-to-date.

In Chapter 2, we introduce a framework of continually pre-training models from ever-changing audio-video data distributions. We propose a method for selecting core audio-video patches from the current task. We utilize the AVM module, which employs cross-attention to calculate importance scores and multimodal correlation scores, and perform probabilistic patch selection based on these scores. This approach identifies semantically intertwined audio-video patches to pre-train target models effectively and make efficient use of GPU memory and limited rehearsal memory space. Moreover, it effectively resolves the issue of multimodal correlation overwriting during continual pre-training.

In Chapter 3, we propose a cluster-level visual instruction tuning data selection for efficient fine-tuning of Large Vision-Language Models (LVLMs). We demonstrate that clustering based on inner activations from a small model can represent visual-linguistic concept-skill compositions shared among diverse tasks in visual instruction tuning datasets. Our method selects more samples from more transferable and less dense clusters, enhancing training efficacy while preserving the diversity of concept-skill compositions within the coreset to ensure better model generalization ability. Our comprehensive experiments demonstrate that our method achieves performances comparable to a fully fine-tuned model, using only a small subset of the entire dataset and incurring the lowest data selection costs.

In the current era, Large Language Models (LLMs) are rapidly evolving with the latest knowledge and thus new and advanced LLMs are constantly released. Recognizing the importance of LLM performance within LVLMs, we emphasize the need for future research to develop effective strategies for seamlessly updating the LLM components within LVLMs. This will ensure that LVLMs remain current and efficient. This avenue for improvement is a key component of our future research.

Bibliography

[1] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Jinxing Zhou, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, and Yiran Zhong. Audio-visual segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[3] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[4] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. URL https://doi.org/10.48550/arXiv.2310.03744.

[5] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[6] Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. *arXiv preprint arXiv:2402.12501*, 2024. URL https://doi.org/10.48550/arXiv.2402.12501.

[7] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. *arXiv preprint arXiv:2402.11690*, 2024. URL https://doi.org/10.48550/arXiv.2402.11690.

[8] Jaewoo Lee, Jaehong Yoon, Wonjae Kim, Yunji Kim, and Sung Ju Hwang. Stella: Continual audio-video pre-training with spatio-temporal localized alignment. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[9] Wentong Liao, Kai Hu, Michael Ying Yang, and Bodo Rosenhahn. Text to image generation with semantic-spatial aware gan. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] Jaewoong Lee, Sangwon Jang, Jaehyeong Jo, Jaehong Yoon, Yunji Kim, Jin-Hwa Kim, Jung-Woo Ha, and Sung Ju Hwang. Text-conditioned sampling framework for text-to-image generation with masked generative models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[11] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable

length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

[12] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[14] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[15] Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. Cross-media structured common space for multimedia event extraction. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2020.

[16] Rui Yan, Mike Zheng Shou, Yixiao Ge, Jinpeng Wang, Xudong Lin, Guanyu Cai, and Jinhui Tang. Video-text pre-training with learned regions for retrieval. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, 2023.

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[18] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, 2019.

[19] Yuan Gong, Andrew Rouditchenko, Alexander H. Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James R. Glass. Contrastive audio-visual masked autoencoder. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[20] Zineng Tang, Jaemin Cho, Yixin Nie, and Mohit Bansal. TVLT: textless vision-language transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, Haoqi Fan, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, and Christoph Feichtenhofer. Mavil: Masked audio-video learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[22] Yan-Bo Lin, Yi-Lin Sung, Jie Lei, Mohit Bansal, and Gedas Bertasius. Vision transformers are parameter-efficient audio-visual learners. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[24] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas M. Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[25] Xixi Hu, Ziyang Chen, and Andrew Owens. Mix and localize: Localizing sound sources in mixtures. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] Xian Liu, Rui Qian, Hang Zhou, Di Hu, Weiyao Lin, Ziwei Liu, Bolei Zhou, and Xiaowei Zhou. Visual sound localization in the wild by cross-modal interference erasing. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, 2022.

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016. URL http://arxiv.org/abs/1612.00796.

[29] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[31] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[32] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[33] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[34] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[35] Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia C. Passaro, Vincenzo Lomonaco, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*, 2022. URL https://doi.org/10.48550/arXiv.2205.09357.

[36] Enrico Fini, Victor G. Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[37] Jaehong Yoon, Sung Ju Hwang, and Yue Cao. Continual learners are incremental model generalizers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

[38] Shipeng Yan, Lanqing Hong, Hang Xu, Jianhua Han, Tinne Tuytelaars, Zhenguo Li, and Xuming He. Generative negative text replay for continual vision-language pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[39] Weiguo Pian, Shentong Mo, Yunhui Guo, and Yapeng Tian. Audio-visual class-incremental learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[40] Shentong Mo, Weiguo Pian, and Yapeng Tian. Class-incremental grouping network for continual audio-visual learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[41] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[42] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[43] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[45] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[46] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2017.

[47] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *arXiv preprint arXiv:1908.04742*, 2019. URL http://arxiv.org/abs/1908.04742.

[48] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[49] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[50] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 1985.

[51] Sunil Hwang, Jaehong Yoon, Youngwan Lee, and Sung Ju Hwang. Efficient video representation learning via masked video modeling with motion-centric token selection. *arXiv preprint arXiv:2211.10636*, 2022. URL https://doi.org/10.48550/arXiv.2211.10636.

[52] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. Unified multisensory perception: Weakly-supervised audio-visual video parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[53] 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.

[54] Vishaal Udandarao. Understanding and fixing the modality gap in vision-language models. *PhD thesis, Master's thesis, University of Cambridge*, 2022.

[55] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[56] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. URL https://doi.org/10.48550/arXiv.2304.10592.

[57] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[58] 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 Marina Meila and Tong Zhang, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[59] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[60] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal LLM. *arXiv preprint arXiv:2312.06742*, 2023. URL https://doi.org/10.48550/arXiv.2312.06742.

[61] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024. URL https://arxiv.org/abs/2404.06512.

[62] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye1, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo,

Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. URL https://arxiv.org/html/2404.16821v2.

[63] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. URL https://doi.org/10.48550/arXiv.2403.18814.

[64] Anthony Meng Huat Tiong, Junqi Zhao, Boyang Li, Junnan Li, Steven C. H. Hoi, and Caiming Xiong. What are we measuring when we evaluate large vision-language models? an analysis of latent factors and biases. In *North American Chapter of the Association for Computational Linguistics*, 2024.

[65] Zhiheng Xi, Rui Zheng, Yuansen Zhang, Xuanjing Huang, Zhongyu Wei, Minlong Peng, Mingming Sun, Qi Zhang, and Tao Gui. Connectivity patterns are task embeddings. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2023.

[66] Oleksiy Ostapenko, Zhan Su, Edoardo Maria Ponti, Laurent Charlin, Nicolas Le Roux, Matheus Pereira, Lucas Caccia, and Alessandro Sordoni. Towards modular llms by building and reusing a library of loras. *arXiv preprint arXiv:2405.11157*, 2024. URL https://arxiv.org/abs/2405.11157.

[67] Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023. URL https://doi.org/10.48550/arXiv.2309.04564.

[68] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[69] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training A better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023. URL https://doi.org/10.48550/arXiv.2307.08701.

[70] Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruction tuning. *CoRR*, abs/2311.15653, 2023. URL https://doi.org/10.48550/arXiv.2311.15653.

[71] Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430*, 2024. URL https://doi.org/10.48550/arXiv.2402.10430.

[72] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024. URL https://doi.org/10.48550/arXiv.2402.04333.

[73] Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. Less is more: Data value estimation for visual instruction tuning. *arXiv preprint arXiv:2403.09559*, 2024. URL https://doi.org/10.48550/arXiv.2403.09559.

[74] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023. URL https://doi.org/10.48550/arXiv.2312.15685.

[75] Kai Wei, Rishabh K. Iyer, and Jeff A. Bilmes. Submodularity in data subset selection and active learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[76] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[77] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[78] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[79] Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[80] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022.

[81] Yu Yang, Siddhartha Mishra, Jeffrey N. Chiang, and Baharan Mirzasoleiman. Smalltolarge (S2L): scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *arXiv preprint arXiv:2403.07384*, 2024. URL https://doi.org/10.48550/arXiv.2403.07384.

[82] Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*, 2023. URL https://doi.org/10.48550/arXiv.2311.08182.

[83] Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv preprint arXiv:2308.12067*, 2023. URL https://doi.org/10.48550/arXiv.2308.12067.

[84] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[85] Thomas FEL, Victor Boutin, Louis Béthune, Remi Cadene, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and

concept importance estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[86] Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020. doi: 10.1073/pnas.1907367117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1907367117.

[87] Matthew Kowal, Richard P Wildes, and Konstantinos G Derpanis. Visual concept connectome (vcc): Open world concept discovery and their interlayer connections in deep models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[88] Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[89] Tanmay Gupta, Kevin Shih, Saurabh Singh, and Derek Hoiem. Aligned image-word representations improve inductive transfer across vision-language tasks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

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

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

[92] Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. Multimodal neurons in pretrained text-only transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023*, 2023.

[93] Haowen Pan, Yixin Cao, Xiaozhi Wang, and Xun Yang. Finding and editing multi-modal neurons in pre-trained transformer. *arXiv preprint arXiv:2311.07470*, 2023. URL https://doi.org/10.48550/arXiv.2311.07470.

[94] Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[95] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. URL https://doi.org/10.48550/arXiv.2402.14289.

[96] Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024. URL https://doi.org/10.48550/arXiv.2402.17762.

[97] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[98] Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *arXiv Preprint 1904.03292*, 2020.

[99] Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[100] Been Kim, Oluwasanmi Koyejo, and Rajiv Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[101] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[102] 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 International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[103] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[104] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[105] Pan Lu, Swaroop Mishra, Tanglin 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.

[106] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[108] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. URL https://arxiv.org/abs/2306.13394.

[109] 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? *arXiv preprint arXiv:2307.06281*, 2023. URL https://doi.org/10.48550/arXiv.2307.06281.

[110] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. URL https://doi.org/10.48550/arXiv.2308.02490.

[111] Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. URL https://doi.org/10.48550/arXiv.2303.09540.

[112] Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *arXiv preprint arXiv:2310.07931*, 2023. URL https://doi.org/10.48550/arXiv.2310.07931.

[113] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[114] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[115] Gabriela Ben Melech Stan, Raanan Y. Yehezkel Rohekar, Yaniv Gurwicz, Matthew Lyle Olson, Anahita Bhiwandiwalla, Estelle Aflalo, Chenfei Wu, Nan Duan, Shao-Yen Tseng, and Vasudev Lal. Lvlm-intrepret: An interpretability tool for large vision-language models. *arXiv preprint arXiv:2404.03118*, 2024. URL https://doi.org/10.48550/arXiv.2404.03118.

Acknowledgment

먼저 부족한 저를 지도해주시고 이끌어주신 황성주 교수님께 감사의 말을 전합니다. 저의 성장을 위한 최고의 환경을 만들어주신 덕분에 석사 기간 동안 많은 성장을 할 수 있었습니다. 또한 저의 진로 고민에도 긴 시간동안 귀 기울여주시고 아낌없는 조언을 주셔 정말 감사드립니다. 심사위원으로 참석해 주신 이주호 교수님과 윤세영 교수님께도 감사를 표합니다.

석사 과정동안 연구에 많은 도움을 준 윤재홍 형, 조재형 형에게 감사의 말을 전합니다. 그리고 김민선 누나, 김동기 형, 백진헌 형, 장상원 형, 박건, 김형도, 안소현, 김강산, 서민주, 그리고 Simon Aytes 덕분에 즐거운 시간을 보낼 수 있었습니다. 힘들 때마다 위로 해주고 고민을 들어주었던 김정환 형, 고준호, 그리고 이다은에게 고개숙여 감사를 표합니다.

그리고 항상 저를 믿고 제 선택을 존중하고 응원해주신 아버지, 어머니, 그리고 누나 동생. 그들의 조건 없는 지지 덕분에 지금까지 용기를 내어 도전할 수 있었습니다. 감사합니다. 사랑합니다.

Curriculum Vitae

Name : Jaewoo Lee

Date of Birth : August 19, 1998

Educations

2023. 3. – Present M.S. in Artificial Intelligence, KAIST

2020. 3. – 2023. 2. B.S. in Electrical Engineering, KAIST

Publications

1. **Jaewoo Lee**, Boyang Li, Sung Ju Hwang. Concept-skill Transferability-based Data Selection for Large Vision-Language Models. Submitted to EMNLP 2024.
2. **Jaewoo Lee***, Jaehong Yoon*, Wonjae Kim, Yunji Kim, Sung Ju Hwang. STELLA: Continual Audio-Video Pre-training with Spatio-Temporal Localized Alignment. International Conference on Machine Learning **(ICML), 2024**. (*: equal contribution)
3. Wonjun Yi, Jung-Woo Choi, **Jaewoo Lee**. Sound-based drone fault classification using multitask learning. The 29th International Congress on Sound and Vibration **(ICSV29)**.