Supplementary for Mode Approximation Makes Good Multimodal Prompts

Anonymous Author(s)
Affiliation
Address
email

In order to provide a comprehensive demonstration of our approach, we will supplement additional details in the Appendix. The arrangement of these sections is as follows: First, we demonstrate the core concepts of our Aurora for clarity. Second, we comprehensively make overall comparisons with existing methods on various multimodal tasks. Then, we provide details regarding the datasets and baselines in Section C, while the concrete training details are outlined in Section D. We then conduct a comprehensive analysis of the computational costs, including time and memory consumption, along with algorithmic complexity in Section E. Furthermore, we provide theoretical support for our approach in Section F. Finally, we present visualizations of our proposed Aurora for several cross-modal tasks to facilitate qualitative comparisons in Section G. To represent our method clearly and concisely, we use lowercase letters for scalars, bold lowercase letters for **vectors**, italicized uppercase letters for *MATRICES*, and bold italicized uppercase letters for ***TENSORS*** in the equations, respectively.

A Concept Definition

According to [1, 27], we will offer a precise definition of the fundamental notions underpinning our key Mode Approximation component.

First, the definition of tensors can be demonstrated as follows:

Definition 1 *(Tensor). Let $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N \in \mathbb{N}$ denote index upper bounds, a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{\mathcal{D}_1 \times \cdots \times \mathcal{D}_N}$ of order N is an N-way array where elements $\mathcal{W}_{d_1,d_2,\cdots,d_n}$ are indexed by $d_n \in \{1, 2, \cdots, \mathcal{D}_n\}$, for $1 \leq n \leq N$.*

Then, the concept of the mode is formulated as follows:

Definition 2 *(Mode). Let $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-dimensional tensor. The mode-k matricization of $\boldsymbol{\mathcal{W}}$, denoted as $\mathcal{W}^{(k)} \in \mathbb{R}^{n_k \times (n_1 \cdots n_{k-1} n_{k+1} \cdots n_d)}$, is obtained by unfolding the tensor along its k-th mode and arranging the entries as rows in a matrix.*

Given the mathematical definition of the mode, we can implement decomposition in the context of CP decomposition. We stack all the weight matrices in the attention layer (i.e., W_q, W_k, W_v) of all the branches into a tensor, which needs to be updated as $\boldsymbol{\Delta\mathcal{W}}$. Since our method assumes that the stack of weight matrices is a three-order tensor, k is three in our settings, and thus the CP decomposition can be illustrated as follows:

$$\boldsymbol{\Delta\mathcal{W}} = \sum_{r=1}^{R} \lambda_r \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{p}_r, \tag{1}$$

where R is the rank, λ_r are non-negative scalar weights, and $\mathbf{u}_r \in \mathbb{R}^{n_1}$, $\mathbf{v}_r \in \mathbb{R}^{n_2}$, $\mathbf{p}_r \in \mathbb{R}^{n_3}$ are non-zero factor vectors. And the mode-k unfolding of the tensor $\boldsymbol{\Delta\mathcal{W}}$ is U, V, P respectively.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

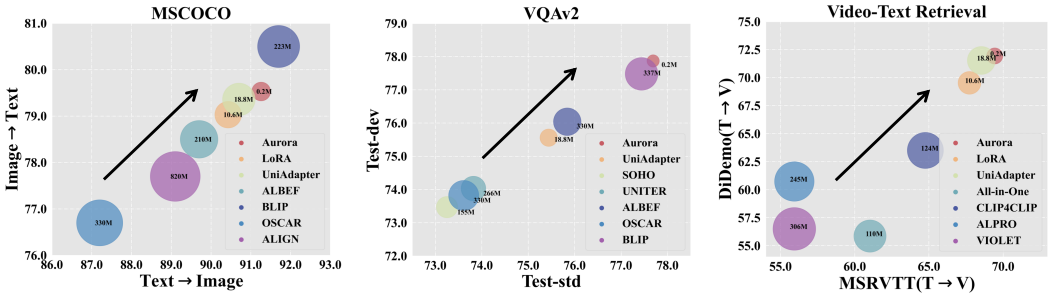

Figure 1: Performance-Parameter comparison of different methods on each multimodal downstream task. Note that the bubble size denotes the size of total tunable parameters.

Our proposed Aurora aims to approximate the latent mode matrix with randomly initialized learnable parameters, which can learn knowledge on downstream tasks in a lightweight manner.

B Overall Comparison

We compare all the baselines for three downstream tasks and presented a comprehensive illustration in Figure 1. The arrow in the figure points towards better performance on dual metrics, as it moves towards the upper right corner. We rank the size of the model parameters and use it as a basis for determining the size of the bubbles, which are also displayed in Figure 1. It is evident that our method performs remarkably well even with smaller parameter sizes, and in several instances, outperforms the fully fine-tuned approach, demonstrating the strength of our architecture.

C Detailed Descriptions for the Baselines & Datasets.

Baselines. For Frozen Backbone methods, UniAdapter [25] is currently the state-of-the-art method for parameter-efficient transfer learning in the field of multimodality and can be considered as a representative of the Adapter class of methods in the multimodal domain. LoRA [14] is another important branch of parameter-efficient transfer learning methods. Its core idea is to use low-rank decomposed matrices to calculate the incremental change of model parameters during adaptation on downstream tasks. To enable comparison with a wider range of baselines, we replicate many of the settings from prior works and reuse their experiment results whenever possible. It should be noted that this means some baselines only appear in specific experiments.

As for Full Fine Tuning methods, we apply UNITER [6], VILLA [12], OSCAR [23], ALIGN [16], ALBEF [22] and BLIP [21] for image-text retrieval tasks, then we use ClipBERT [19], Frozen in Time [3], ALPRO [20], VIOLET [11], All-in-one [29], CLIP4Clip [26] and CLIP-Hhiker [4] for text-video retrieval tasks, finally we adopt ClipBERT[19], ALPRO [20], Just-Ask [32], VIOLET [11], MERLOT [33], All-in-one [29] for VideoQA task while adopt VL-T5/BART [7], SOHO [15], OSCAR [23], UNITER [6], ALBEF [22] and BLIP [21] for VQA tasks.

Datasets. We provide a comprehensive introduction to the datasets of various downstream tasks in the multimodal scenario, as outlined below:

- **MSCOCO** [24] is a large scale image-text dataset and each image is annotated with five captions. Following [25, 17], we use Karpathy split of MSCOCO: 5,000 images for testing, 5,000 images for validation, and the rest for training.
- **Flickr30K** [28] contains 31,000 images collected from Flickr. Each image is usually referenced with five human annotations. Following previous works [25, 10], we use 1,000 images for testing, another 1000 for validation, and the rest for training.
- **MSR-VTT** [31] contains 10,000 video clips and each video clip is annotated with 20 English sentences. Following recent works [25, 26], we adopt 1k-test split for training and testing.
- **DiDemo** [2] is one of the most commonly used datasets for the temporal localization of events in videos. It contains about 10,000 videos and 40,000 annotations. we follow [25, 3] to concatenate all

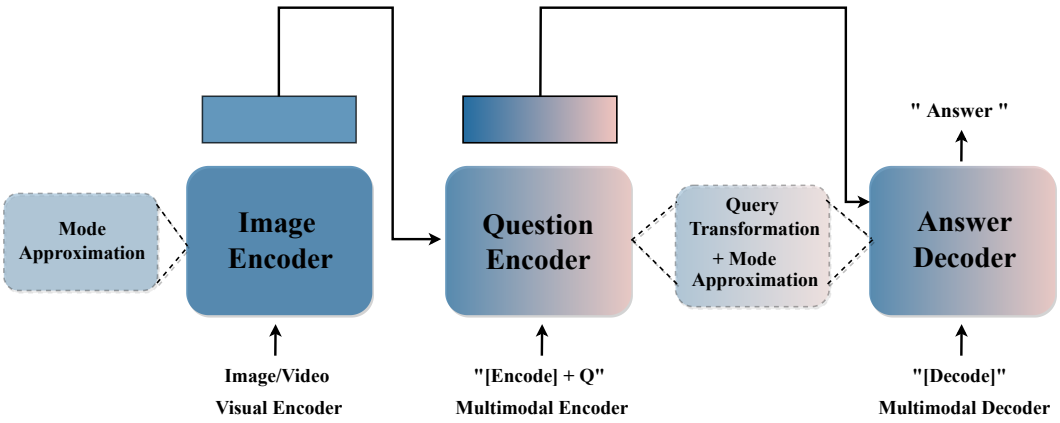

Figure 2: Model architecture for the Visual Question Answering tasks on both images and videos.

descriptions corresponding to the same video into a single sentence to conduct actually paragraph-to-video retrieval task.

- **VQAv2** [13] is one of the most famous visual question answering datasets which contains 83k/41k/81k images for training/validation/testing. Following [25, 22, 21], we use both training and validation splits of VQAv2 and additional training samples from Visual Genome [18] for training. The results should be evaluated by the official server and we report the results on the test-dev and test-std splits.
- **MSRVTT-QA** [30] is one of the most popular video question answering datasets. It's constructed based on MSRVTT and has 243k open-ended questions associated with 10k videos. We follow [25, 19] to employ the standard split for training and testing.

D Training Details

D.1 Frozen Backbone

BLIP [21] is a unified VLP framework which has multimodal mixture of encoder-decoder(MED) architecture with both understanding and generation capabilities. In our experiments, we utilize BLIP-base as the frozen backbone and the pre-trained weights can be downloaded from Salesforce. Its visual encoder is ViT-B [9] and the text encoder is the same as BERT [8] while the text decoder replaces the self-attention layers with causal self-attention layers. It uses cross-attention layers to gather information from encoded visual representations using the textual representations as query. It's flexible to choose different components in the BLIP architecture to perform different multimodality downstream tasks.

Another important design in BLIP is CapFlit, It contains a Captioner to generate captions given web-searched images and a Filter to remove noisy image-text pairs, both Captioner and Filter are finetuned individually on the COCO dataset while using different objective loss. In addition, BLIP uses momentum technology to further improve the correctness of the image-text matching relationship.

D.2 Architecture for VQA Tasks

Figure 2 shows the architecture of Aurora for Visual Question Answering tasks. Compared to Retrieval tasks, the VQA architecture has an additional answer decoder. During fine-tuning, the images/videos are first encoded by a single-modal visual encoder, and then the image/video-text pairs are fused using a multimodal encoder and given to the decoder for prediction. Answers are used as ground truth and Language-Model Loss is utilized for parameter updating throughout the entire training process. As the ITM Loss is no longer needed, we remove the Informative Context Enhancement module from the VQA architecture. Meanwhile, we retain the Gated Query Transformation module to preserve the complete semantic information of the question representations as much as possible. Finally, in order to further reduce the number of parameters, we share the learnable parameters of the multimodal encoder and the multimodal decoder.

Table 1: Training time and GPU memory comparison.

Method	#Tunable	MSCOCO		FLICKR30K		MSRVTT-QA		VQAv2		DiDemo		MSRVTT	
		Time	Memory	Time	Memory	Time	Memory	Time	Memory	Time	Memory	Time	Memory
UniAdapter (r=512)	18.8M	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
UniAdapter (r=128)	4.6M	0.86	0.95	0.93	0.95	0.89	0.91	0.79	0.92	**0.88**	0.94	**0.94**	0.94
Aurora (r=128)	0.2M	0.90	**0.93**	0.93	**0.94**	0.88	**0.89**	0.79	0.91	0.90	0.92	0.95	0.93
Aurora (r=64)	0.1M	**0.84**	**0.93**	**0.92**	**0.94**	**0.86**	**0.89**	**0.76**	**0.89**	**0.88**	**0.88**	**0.94**	**0.92**

D.3 Implementation Details

In this section, we give more training details about our Aurora.

- For image-text downstream tasks, we set the image size into 384×384. We use cosine decay to update the learning rate during training. We set the batch size to 16 for each GPU and train a total of 6 epochs.
- For video-text retrieval tasks, we randomly sample T = 8(16) frames for each video during training(testing) while setting the frame size to be 224×224. We use cosine decay to update the learning rate, we set the batch size to 8 per GPU during training and train for a total of 5 epochs.
- For VideoQA task, we also sample 8 frames per video for training while the frame size is changed to 384×384. While training, we set the batch size to 4 per GPU and train 10 epochs in total. During the evaluation, we randomly sample 16 frames for each video, and we use greedy search to generate the next token when producing answer for its corresponding question.
- For VQA task, we set the image size to 480×480 for training/inference and adopt a batch size of 16 for each GPU. We also adopt cosine decay to change the value of learning rate at different epochs and we train 10 epochs.

We also perform a simple cleaning of the text data and truncate all words beyond the maximum length of the sentence. All data processing and partitioning are consistent with UniAdapter and LoRA to ensure fair comparison. When implementing CP decomposition, textual encoder, visual encoder, and multimodal encoder share the same global mode factor U and factor V to do parameter sharing and we initialize the weights of factor $V$to be zero.

E Cost Analysis

In Table 1, following [25], we report the training time and GPU memory cost for both retrieval and VQA tasks. We regard the training time and memory cost of UniAdapter(r=512) as one unit. Since we adopt the same backbone models, the forward propagation process of the two methods, Aurora and UniAdapter, is almost consistent, and the time cost is similar. However, our Aurora has fewer trainable parameters, resulting in a slightly smaller GPU memory footprint.

Then, We will give a theoretical analysis of the parametric complexity of the three PETL methods. Assume that the frozen backbone's visual, textual, and multimodal encoder all have L transformer layers while multimodal encoders contain cross-attention modules and visual, textual encoders contain self-attention modules. We only approximate the Query/Key/Value weight matrix in these attention-based modules. Let d donates the dimension of the hidden feature and r for rank, so the parametric complexity of the LoRA is L × (3 + 6) × 2dr $\sim \mathcal{O}$(Ldr), the complexity of the UniAdapter is L × 4dr $\sim \mathcal{O}$(Ldr), and the complexity of our Aurora is L × (3 + 6) × 2r + 2dr + 2Ld $\sim \mathcal{O}$((L+d)r), normally r and L are much smaller than d, so from the above analysis we can draw the conclusion that when r increases, our Aurora can achieve the lowest parameter cost.

F Theoretical Analysis

Write $\Delta\boldsymbol{\mathcal{W}} = \sum_{r=1}^{R} \lambda_r(\boldsymbol{u_r} \circ \boldsymbol{v_r} \circ \boldsymbol{p_r})$, and the (i, j, k)-element of $\Delta\boldsymbol{\mathcal{W}}$ is:

$$\Delta\boldsymbol{\mathcal{W}}_{ijk} = \sum_{r=1}^{R} \lambda_r u_{ri} v_{rj} p_{rk}. \tag{2}$$

Recall that the *Frobenius norm* of a tensor $\mathcal{X} \in \mathbb{R}^{d\times d\times N}$ is given by:

$$\|\mathcal{X}\|_F = \left(\sum_{i_1=1}^{d}\sum_{i_2=1}^{d}\sum_{i_3=1}^{N} |x_{i_1,i_2,i_3}|^2\right)^{1/2}, \tag{3}$$

Hence, it suffices to analyze the convergence rate of our parameter tensors in the Euclidean space $\mathbb{R}^{ddN}$, where ddN denotes the product of $d \times d \times N$ in order to distinguishing from the space of multi-dimensional arrays of size $d \times d \times N$.

Assumption F.1 *We identify $\mathbb{R}^{d\times d\times N}$ with $\mathbb{R}^{ddN}$, and let the loss function $\mathcal{L}$ be defined on $\mathbb{R}^{ddN}$, while we still write $\mathcal{L}(\boldsymbol{\mathcal{W}})$ where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d\times d\times N}$ is a tensor. We will also use Frobenius norm and ℓ^2 norm interchangeably, which means that:*

$$\|\mathcal{X}\|_F = \|\mathcal{X}\|_2. \tag{4}$$

Assumption F.2 *We assume that the loss function $\mathcal{L} : \mathbb{R}^{ddN} \to \mathbb{R}$ has the following property:*

1. *$\mathcal{L}$ is injective.*
2. *$\mathcal{L}$ is strongly convex: there exist m and M such that:*

$$mI \preceq \nabla^2\mathcal{L}(\mathcal{X}) \preceq MI. \tag{5}$$

That is, $\nabla^2\mathcal{L}(\mathcal{X}) - mI$ is positive semidefinite and $\nabla^2\mathcal{L}(\mathcal{X}) - MI$ is negative semidefinite.

Let $\{\lambda_r^{(0)}, \boldsymbol{u}_r^{(0)}, \boldsymbol{v}_r^{(0)}, \boldsymbol{p}_r^{(0)} : r = 1, \cdots, R\}$ be the randomly initialized vectors used for tensor decomposition, where $\lambda_r^{(0)} \in \mathbb{R}, \boldsymbol{u}_r^{(0)} \in \mathbb{R}^d, \boldsymbol{v}_r^{(0)} \in \mathbb{R}^d, \boldsymbol{p}_r^{(0)} \in \mathbb{R}^N$. Denote

$$\Delta\boldsymbol{\mathcal{W}}^{(0)} = \boldsymbol{\mathcal{W}}_0 = \sum_{r=1}^{R} \lambda_r^{(0)} \boldsymbol{u}_r^{(0)} \circ \boldsymbol{v}_r^{(0)} \circ \boldsymbol{p}_r^{(0)}. \tag{6}$$

Let $\Delta\boldsymbol{\mathcal{W}}^{(n)}$ be the parameter tensor returned by the nth training epoch, that is,

$$\Delta\boldsymbol{\mathcal{W}}^{(n)} = \sum_{r=1}^{R} \lambda_r^{(n)} \boldsymbol{u}_r^{(n)} \circ \boldsymbol{v}_r^{(n)} \circ \boldsymbol{p}_r^{(n)}. \tag{7}$$

Theorem F.1 *Under the above assumptions, and suppose that we train for n epochs with $\eta \leq 1/M$ using gradient descent. Let $\boldsymbol{\mathcal{W}}^*$ be the optimal parameter tensor, then,*

$$\mathcal{L}\left(\boldsymbol{\mathcal{W}}_0 + \Delta\boldsymbol{\mathcal{W}}^{(n)}\right) \to \mathcal{L}\left(\boldsymbol{\mathcal{W}}^*\right) \quad (n \to \infty). \tag{8}$$

Moreover, $\boldsymbol{\mathcal{W}}^*$ is unique.

Proof F.1 *The proof follows from [5]. For notational convenience let $\mathcal{X}^{(n)} = \boldsymbol{\mathcal{W}}_0 + \Delta\boldsymbol{\mathcal{W}}^{(n)}$, and denote the optimal value $\mathcal{L}(\boldsymbol{\mathcal{W}}^*)$ by λ^*. We will begin by analyzing the convergence using arbitrary $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{ddN}$, and then plug in our parameter tensors. By Taylor's theorem we can write:*

$$\mathcal{L}(\mathcal{Y}) = \mathcal{L}(\mathcal{X}) + \nabla\mathcal{L}(\mathcal{X})^T(\mathcal{Y} - \mathcal{X}) + \frac{1}{2}(\mathcal{Y} - \mathcal{X})^T\nabla^2\mathcal{L}(\mathcal{Z})(\mathcal{Y} - \mathcal{X}), \tag{9}$$

where $\mathcal{Z}$ lies in the line segment joining $\mathcal{X}$ and $\mathcal{Y}$. By the strong convexity assumption, we have,

$$\frac{1}{2}(\mathcal{Y} - \mathcal{X})^T\nabla^2\mathcal{L}(\mathcal{Z})(\mathcal{Y} - \mathcal{X}) \geq \frac{1}{2}(\mathcal{Y} - \mathcal{X})^T m(\mathcal{Y} - \mathcal{X}) = \frac{m}{2}\|\mathcal{Y} - \mathcal{X}\|_2^2. \tag{10}$$

Hence

$$\mathcal{L}(\mathcal{Y}) \geq \mathcal{L}(\mathcal{X}) + \nabla\mathcal{L}(\mathcal{X})^T(\mathcal{Y} - \mathcal{X}) + \frac{m}{2}\|\mathcal{Y} - \mathcal{X}\|_2^2. \tag{11}$$

Now we use $\|\nabla\mathcal{L}(\mathcal{X})\|_2$ to bound $\mathcal{L}(\mathcal{X}) - \lambda^*$. The right-hand side of (11) is a convex quadratic function of $\mathcal{Y}$, hence $\mathcal{Y}^* = \mathcal{X} - 1/m\nabla\mathcal{L}(\mathcal{X})$ is the minimizer, thus,

$$\mathcal{L}(\mathcal{Y}) \geq \mathcal{L}(\mathcal{X}) + \nabla\mathcal{L}(\mathcal{X})^T(\mathcal{Y}^* - \mathcal{X}) + \frac{m}{2}\|Y^* - \mathcal{X}\|_2^2 \tag{12}$$

$$= \mathcal{L}(\mathcal{X}) + \nabla\mathcal{L}(\mathcal{X})^T\left(-\frac{1}{m}\nabla\mathcal{L}(\mathcal{X})\right) + \frac{m}{2}\left\|\frac{1}{m}\nabla\mathcal{L}(X)\right\|_2^2 \tag{13}$$

$$= \mathcal{L}(\mathcal{X}) - \frac{1}{2m}\|\nabla\mathcal{L}(\mathcal{X})\|_2^2. \tag{14}$$

Since $\mathcal{Y}$ is arbitrary, plugging $\mathcal{X} = \mathcal{X}^{(n)}$, we have

$$\lambda^* \geq \mathcal{L}(\mathcal{X}^{(n)}) - \frac{1}{2m}\left\|\nabla\mathcal{L}(\mathcal{X}^{(n)})\right\|_2^2 \tag{15}$$

By the assumption $\nabla^2\mathcal{L}(\mathcal{X}) \preceq MI$ we have

$$\mathcal{L}(\mathcal{Y}) \leq \mathcal{L}(\mathcal{X}) + \nabla\mathcal{L}(\mathcal{X})^T(\mathcal{Y} - \mathcal{X}) + \frac{M}{2}\|\mathcal{Y} - \mathcal{X}\|_2^2. \tag{16}$$

Plugging in $\mathcal{Y} = \mathcal{X}^{(n)} - \eta\nabla\mathcal{L}(\mathcal{X}^{(n)})$ yields

$$\widetilde{\mathcal{L}}(t) \leq \mathcal{L}(\mathcal{X}^{(n)}) - \eta\|\nabla\mathcal{L}(\mathcal{X})\|_2^2 + \frac{M\eta^2}{2}\|\nabla\mathcal{L}(\mathcal{X})\|_2^2. \tag{17}$$

Now we minimize over η on both sides of (17), and denote the optimal value by $\widetilde{\mathcal{L}}(\eta^*)$. The right-hand side of (17) is simple quadratic, hence it is minimized by $\eta = 1/M$, and

$$\min(\text{RHS}) = \mathcal{L}(\mathcal{X}) - \frac{1}{2M}\|\nabla\mathcal{L}(\mathcal{X})\|_2^2. \tag{18}$$

Now,

$$\mathcal{L}(\mathcal{X}^{(n)} + \eta\Delta\mathcal{X}^{(n)}) = \widetilde{\mathcal{L}}(t^*) \leq \mathcal{L}(\mathcal{X}^{(n)}) - \frac{1}{2M}\|\nabla\mathcal{L}(\mathcal{X})\|_2^2. \tag{19}$$

Subtracting λ^* on both sides, we have,

$$\mathcal{L}\left(\mathcal{X}^{(n)} + \eta\Delta\mathcal{X}^{(n)}\right) - \lambda^* \leq \mathcal{L}(\mathcal{X}^{(n)}) - \lambda^* - \frac{1}{2M}\|\nabla\mathcal{L}(\mathcal{X})\|_2^2, \tag{20}$$

by (15) we have,

$$\mathcal{L}\left(\mathcal{X}^{(n+1)}\right) = \mathcal{L}\left(\mathcal{X}^{(n)} + \eta\Delta\mathcal{X}^{(n)}\right) - \lambda^* \leq \left(1 - \frac{m}{M}\right)\left(\mathcal{L}(\mathcal{X}^{(n)}) - \lambda^*\right). \tag{21}$$

By mathematical induction, we obtain,

$$\mathcal{L}(\mathcal{X}^{(n)}) - \lambda^* \leq \left(1 - \frac{m}{M}\right)^n\left(\mathcal{L}(\mathcal{X}^{(0)} - \lambda^*\right) \to 0 \quad (n \to \infty), \tag{22}$$

therefore $\mathcal{L}(\mathcal{X}^{(n)}) \to \lambda^*$ as $n \to \infty$.

Suppose $\boldsymbol{\mathcal{V}}^*$ is another optimal parameter tensor, then by the same argument we have $\mathcal{L}(\mathcal{X}^{(n)}) \to \mathcal{L}(\boldsymbol{\mathcal{V}}^*)$ as $n \to \infty$. Since $\mathbb{R}^{ddN}$ is a Hausdorff space, $\mathcal{L}(\boldsymbol{\mathcal{V}}^*) = \mathcal{L}(\boldsymbol{\mathcal{W}}^*)$. By our assumption that $\mathcal{L}$ is injective, $\boldsymbol{\mathcal{W}}^* = \boldsymbol{\mathcal{V}}^*$.

G Visualization Analysis

G.1 Parameter Distribution

Figure 3 and Figure 4 show more details on the parameter distribution comparisons between our Aurora and the pre-trained model and the full fine-tuned model, in which similar results can be observed on $\boldsymbol{\mathcal{W}}_k$ and $\boldsymbol{\mathcal{W}}_v$. We can see that the mode approximation parameters adjust the original weights, and change the distribution of weights and biases to fit the downstream task. It can be concluded that Aurora has several advantages over traditional fine-tuning approaches. First, it avoids over-fitting to specific downstream tasks by only adjusting the pre-trained model parameters in a small local range. Second, it reduces the amount of training required on new data, making it more efficient and cost-effective. Last but not least, Aurora can further improve the model's performance on downstream tasks.

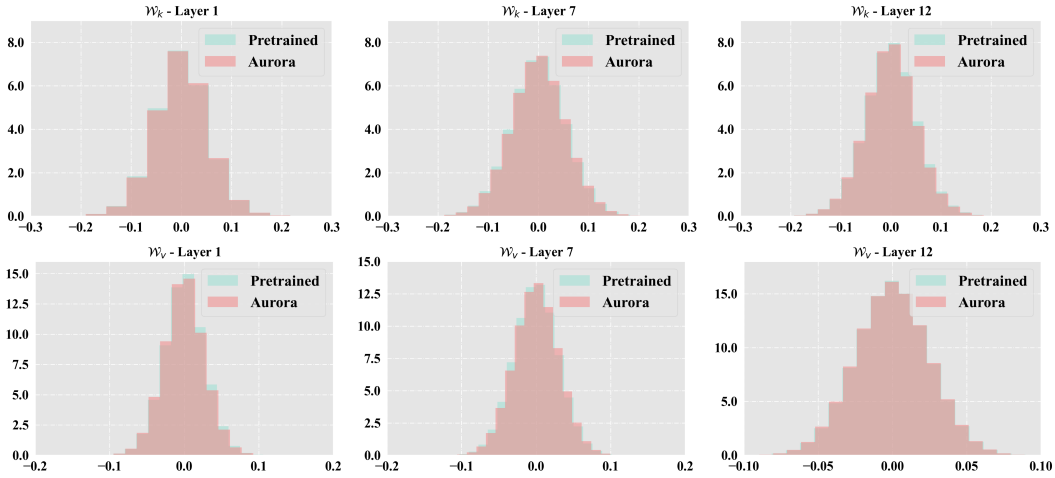

Figure 3: We represent the parameter distribution on different layers of the pre-trained model (BLIP) *vs.* our Aurora, which is tuned on MSCOCO for image-text retrieval. Notably, $\boldsymbol{\mathcal{W}}_k$ and $\boldsymbol{\mathcal{W}}_v$ are the stack of the key and value projection matrices in different modality branches.

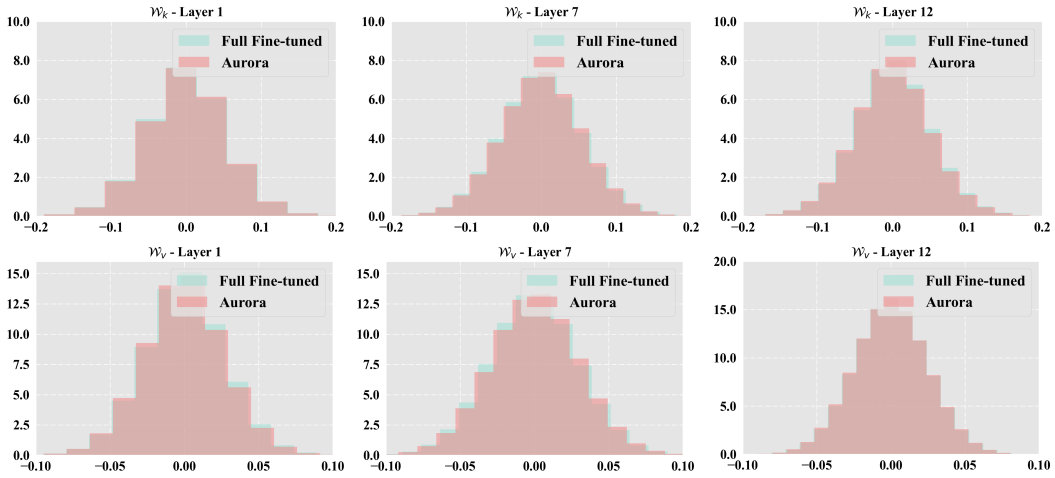

Figure 4: We represent the parameter distribution on different layers of the full fine-tuned model *vs.* our Aurora, which is tuned on MSCOCO for image-text retrieval. Notably, $\boldsymbol{\mathcal{W}}_k$ and $\boldsymbol{\mathcal{W}}_v$ are the stack of the key and value projection matrices in different modality branches.

G.2 Case Study

Visual-Text Retrieval. Figure 5 demonstrates some actual examples of Aurora performing text-to-video task on MSRVTT test set. In conclusion, the results presented highlight the exceptional performance of Aurora in searching relevant videos from textual descriptions. The accuracy and realism of the returned videos demonstrate the effectiveness of our proposed method in understanding the relationship between text and visual content.

Visual Question Answering. Figure 6 gives some question-answering examples of Aurora and UniAdapter on the MSRVTT-QA dataset. Specifically, our method is able to reason about the meaning of the text and video information to answer the questions more accurately than UniAdapter. This is an important result because the ability to reason about the meaning of both text and visual information is essential for understanding multimodal data.

Overall, the qualitative results shown in Figure 5 and Figure 6 demonstrate the effectiveness of our proposed method in both multimodal retrieval and question-answering tasks. We believe that our

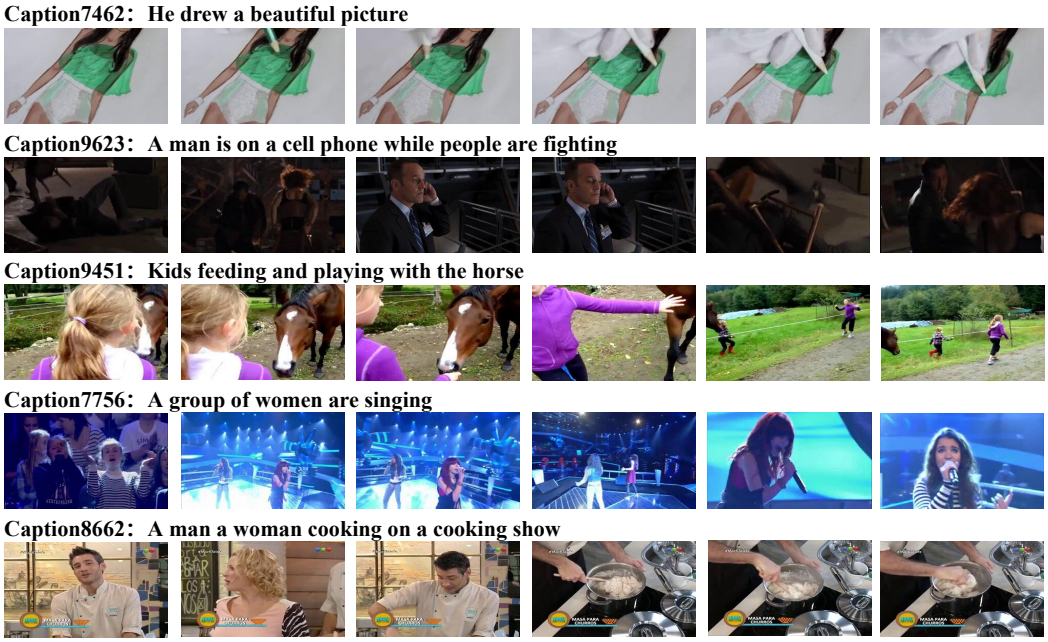

Figure 5: Video-Text retrieval cases on MSRVTT test set.

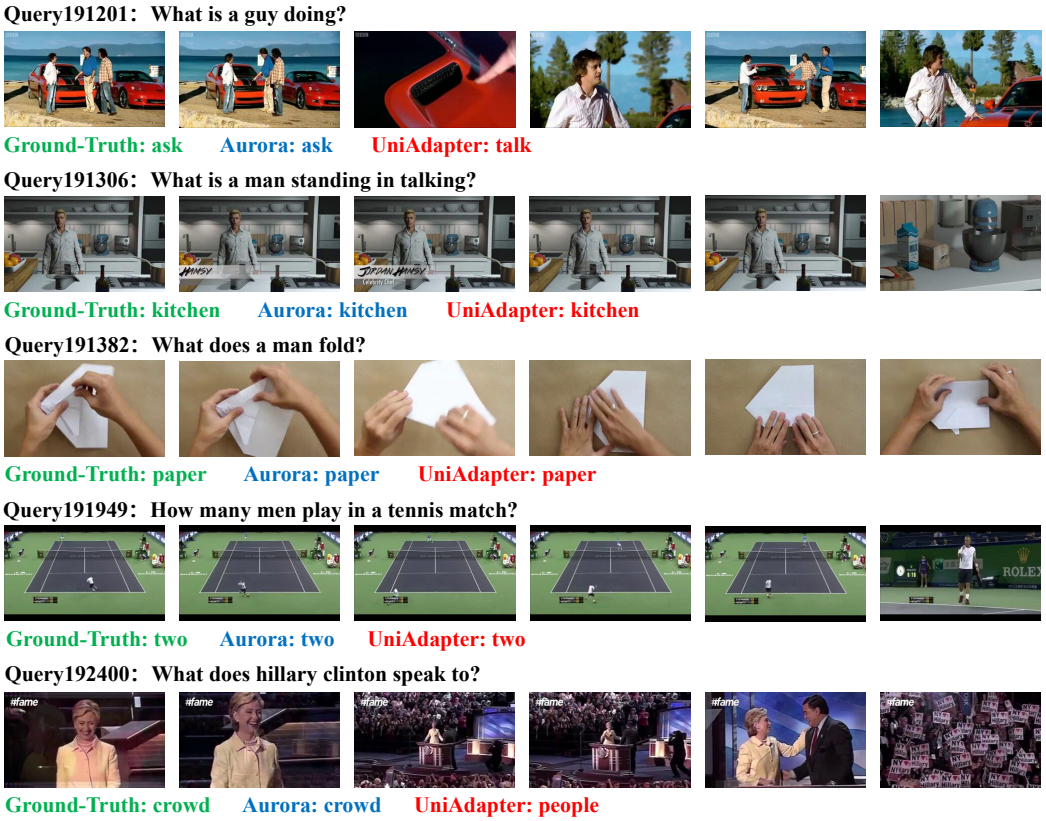

Figure 6: Video Question Answering cases on MSRVTT-QA test set.

approach has the potential to be used in many multimodal applications, where understanding and analyzing multimedia data is essential.

References

[1] RZdunek ACichocki et al. Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation, 2009.

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*, 2022.

[5] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pages 104–120. Springer, 2020.

[7] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[11] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

[12] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[14] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[15] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985, 2021.

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[17] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[19] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021.

[20] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022.

[21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[22] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[23] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[25] Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling. *arXiv preprint arXiv:2302.06605*, 2023.

[26] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[27] Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. *Advances in neural information processing systems*, 32, 2019.

[28] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[29] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022.

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

[31] 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 conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[32] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697, 2021.

[33] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021.