Too Many Tokens, Too Little Focus? Rethinking Multimodal Attention with Soft Masking

Anonymous ACL submission

Abstract

Despite the remarkable success of Transformer-based self-attention in many domains, its effectiveness often diminishes in highly complex multimodal scenarios, where varying token granularities and long, noisy inputs can overwhelm the model. In this paper, we introduce **Soft Token Attention Masking Process (STAMP)**, a novel soft-masking mechanism designed to prioritize the most relevant tokens across visual, audio, and textual streams. By refining attention maps globally, STAMP adapts each token's contribution based on its contextual importance, preserving critical temporal and intermodal cues without discarding valuable information. We integrate STAMP into a multi-layer Transformer pipeline and thoroughly evaluate it on challenging video understanding datasets such as MADv2 and QVHighlights. Experimental results show that STAMP not only delivers significant performance gains but also offers a robust solution for complex multimodal tasks.

1 Introduction

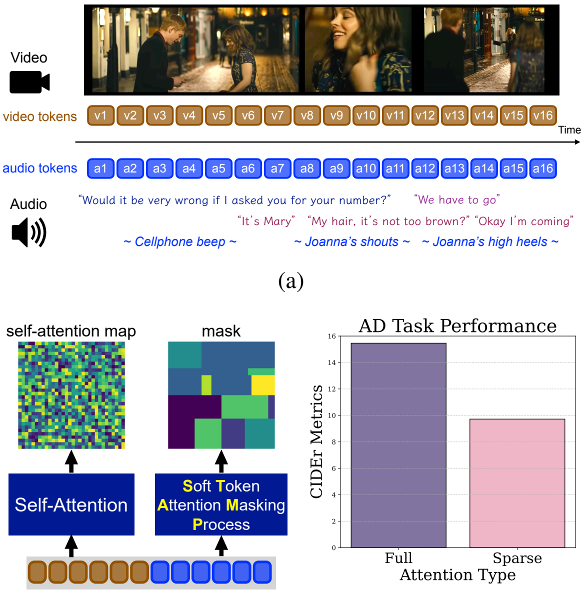
In recent years, there has been a surge of interest in *multimodal* tasks that combine language with other data sources, such as Audio Descriptions in text (Soldan et al., 2022; Han et al., 2023c,b) and video grounding (Moon et al., 2023; Lei et al., 2021; Barrios et al., 2023). Each modality—ranging from visual frames, audio signals, and textual transcripts—provides complementary cues about the same event or scene. However, this richness also introduces complexity, as models must identify and prioritize the *most relevant* portions of each modality to effectively carry out tasks like retrieval or description.

Consider a movie scene composed of hundreds of video frames, aligned audio segments, and partial textual annotations (*e.g.*, dialogue). Although these tokens align in time, each token may also align conceptually with multiple tokens across different modalities, as depicted in Figure 1(a). For example, *"Joanna's shouts"* might not map to a single frame or audio clip but rather a sub-sequence of frames and sound segments. While self-attention mechanisms in Transformers provide a powerful way to learn pairwise relationships among tokens, they can become overwhelmed by *redundant* data. This is especially apparent in highly dynamic or lengthy inputs, where attention maps risk overemphasizing repeated content.

On the one hand, using long token sequences can capture detailed temporal and contextual cues. Conversely, such sequences often introduce excessive redundancy. For instance, consecutive frames may not yield additional insights for an audio description task, yet their mere presence forces the model to allocate attention to irrelevant tokens. Conversely, overly sparse representations (Lin et al., 2022) may jeopardize essential temporal context, making it difficult to grasp the nuances of dynamic content such as movies, vlogs, or news broadcasts. This effect is illustrated in Figure 1(c), which shows that increasing sparsity in the multimodal encoding stage leads to a decline in performance for the Audio Description Task.

While research in NLP has demonstrated *dynamic masking* and other adaptive approaches (Fan et al., 2021; Tang et al., 2021; Lin and Joe, 2023; Rende et al., 2024), these methods remain underexplored in **multimodal encoding stages**. This motivates a mechanism that can selectively filter tokens from multiple modalities without losing crucial information.

In this work, we propose a novel **Soft Token Attention Masking Process (STAMP)** , which dynamically computes a weight matrix to refine attention maps. STAMP acts as a *soft* filtering operation: rather than entirely discarding tokens, it adjusts their relative contribution based on contextual relevance. Here, 'context' spans temporal coherence,



(a)

(b)

(c)

Figure 1: (a) While video and audio tokens naturally align in time, their associations can extend beyond temporal boundaries. For example, "Joanna's shouts" may correspond to multiple video tokens (i.e. not just v8-11, but also v13-16). (b) The Self-Attention module (Vaswani et al., 2017) can capture these attention scores *locally*, token-versus-token. We introduce the **Soft Token Attention Masking Process (STAMP)**, a novel concept that enables a holistic overview of the entire sequence of input tokens, generating a mask that captures attention structures *globally*. (c) When increasing the sparsity in the multimodal encoding stage, the performance for Audio Description Task decreases.

intermodal links, and evolving scene dynamics. By inspecting the entire multimodal sequence, STAMP zeroes in on the tokens that matter most for tasks like multimodal retrieval or audio-visual captioning. Moreover, STAMP seamlessly integrates with standard Transformer Encoders, offering a flexible and generalizable approach for multimodal learning. Moreover, STAMP seamlessly integrates with standard Transformer Encoders, offering a flexible, **plug-and-play** solution that can be easily assembled into any state-of-the-art Transformer architecture for multimodal learning.

In summary, our key **contributions** are threefold:

1. We propose a novel soft-token masking mechanism, STAMP, that adaptively emphasizes important tokens within multimodal inputs.

2. We demonstrate how STAMP enhances feature representations for tasks such as audio description and video grounding, leading to improved downstream performance.

3. We provide comprehensive experiments and analyses demonstrating that STAMP integrates smoothly into various Transformer architectures, supporting different attention mechanisms such as self-attention, cross-attention, and FlashAttention v2 (Dao, 2023). Its adaptability ensures efficient handling of large-scale multimodal data across diverse Transformer models.

2 Related Work

2.1 Multimodal Transformers

A predominant area of prior exploration in aligning multiple modalities centers around contrastive learning, a method extensively utilized in both image-text and video-audio alignment contexts (Chen et al., 2020; Khosla et al., 2020; Radford et al., 2021; He et al., 2019; Han et al., 2023a; Zhang et al., 2023a). Recent investigations have also delved into merging diverse modalities within a unified feature space through the incorporation of cross-attention layers (Chen et al., 2021; Lee et al., 2021; Wei et al., 2020; Moon et al., 2023). Furthermore, there is a growing trend of leveraging Transformer capabilities for multimodal fusion tasks (Luo et al., 2021; Kamath et al., 2021; Han et al., 2023a; Barrios et al., 2023; Lei et al., 2021). Our decision to employ a multimodal transformer in our design is rooted in its unparalleled capability to integrate information across diverse modalities, thus fostering a more comprehensive understanding of the input data. Through the utilization of this unified architecture, we are enabled to effectively capture intricate interactions within the sequence, strategically prioritizing relevant cues based on their significance. In contrast to conventional methodologies that treat modalities in isolation, the multimodal transformer facilitates the seamless integration of contextual information, thereby yielding more coherent and nuanced representations.

2.2 Language Models for Video Description

To adapt a Large Language Model (LLM) for AD generation, we incorporate an adapter module. This module processes audiovisual features and transforms them into the feature space of our LLM. The concept of training an adapter module rather than finetuning the entire LLM to account for a new modality has been widely explored (Yi-Lin Sung, 2022; Hu et al., 2023), but the method most similar to ours is LLaMA-Adapter (Zhang et al., 2023b; Gao et al., 2023). LLaMA adapter, however, does not account for audio data. Our method follows that of LLaMA-Adapter closely, but changes the input feature space to include both audio and video features. The previous State-of-the-Art in our specific task (generating audio descriptions of movie clips) on the MAD dataset are the AutoAD models (Han et al., 2023c,b). We are able to generate comparable results with significantly less fine tuning and contextual information. Recent models have also achieved significant results in finding important moments in longer videos, but these contributions are not particularly relevant to ours because we focus on describing shorter video segments (Lei et al., 2021; Barrios et al., 2023). Another recent result similar to ours is the Video-LLaMA model, which focuses on general purpose visual question answering but uses a Q-Former instead of an adapter module to fuse the visual, audio, and text modalities (Zhang et al., 2023a).

2.3 Masking Attention

In the field of Natural Language Processing, researchers have explored various methods of constructing attention masks, while also investigating their impact on transformer architectures (Fan et al., 2021; Tang et al., 2021; Lin and Joe, 2023; Rende et al., 2024). Conversely, this exploration has received limited attention in Computer Vision (Li et al., 2021; Lin et al., 2022). Motivated by this disparity, our objective is to investigate this phenomenon, particularly in the context of multimodal data, and its implications for task performance. Unlike the approach proposed by SwinBert (Lin et al., 2022), which advocates for a sparse and learnable mask, our focus aligns more closely with the principles of Mask Attention Networks (Fan et al., 2021). Instead of relying on a static mask matrix, which may restrict the model's ability to capture local relationships effectively, we propose employing a Soft Token Attention Masking Process (STAMP). This adaptive mechanism aims to prioritize and regulate attention tokens within long sequences based on their contextual significance in a dynamic manner.

3 Soft Token Attention Masking Process (STAMP)

Our goal is to train a soft token masking mechanism with context awareness, enabling it to dynamically prioritize and adjust token importance based on their relevance within a complex sequence. The term “context" here encompasses multiple dimensions: temporal relationships, intermodal associations and dynamic changes throughout the input sequence. This adaptable mechanism can also be seamlessly incorporated into any of the existing Transformer Encoders. Figure 2 shows the overview of the STAMP architecture.

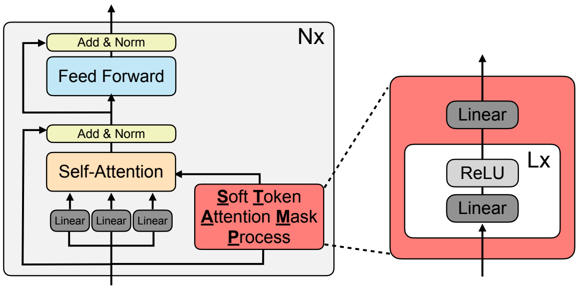


Figure 2: Overview of the Soft Token Attention Masking Process (STAMP). The STAMP module processes the entire input sequence to generate a mask. This mask is applied element-wise (e.g., via addition or multiplication, as detailed in Section 3.3) to modify the attention scores produced by the Transformer Encoders.

3.1 Definition

We aim to generate an attention mask that adeptly prioritizes and regulates tokens according to their importance within a sequence. For this purpose, we develop a learnable module (denoted as $\mathcal{F}$), which receives a sequence of tokens $\mathbf{X}$ as input and returns a mask $\mathbf{M}$, as shown in Equation 1. The shape of the mask $\mathbf{M}$ depends on the sequence length and the purpose of the multi-head attention, whether it is self-attention or cross-attention.

$$\mathcal{F}(\mathbf{X}) \rightarrow \mathbf{M} \tag{1}$$

Attention. In the context of self-attention, the resulting mask output can be represented as (B, N, N), where B is the batch size and N is the number of tokens. For instance, consider a multimodal sequence with 128 tokens and a batch size of 1. In this case, the output of our STAMP would have the dimensions $(1, 128, 128)$. In the cross-attention setup, the mask shape is given by (B, N_q, N_k), where N_q represents the length of the Query tensor and N_k represents the length of the Key tensor. For example, if we perform cross-attention with 75 visual features as the query, 32 text features as the key, and a batch size of 1, the output of STAMP will be $(1, 75, 32)$. To ensure clarity for the reader, all content in the Section 3 refers to the self-attention scenario unless otherwise specified.

Scalability. The mask can be applied either globally across all transformer layers in the stack (Section 3.2) or individually for each layer (Section 3.4). This flexibility allows for different masks to be used at various depths, meaning that each layer can have its own set of learnable parameters, capturing hierarchical information at different levels.

3.2 Soft Token Attention Masking Process (STAMP) Module

The Soft Token Attention Masking Process (STAMP) module processes an input sequence through a stack of linear layers with ReLU activations. Let $\mathbf{X} \in \mathbb{R}^{B\times N\times D}$ denote the input to the STAMP module, where B is the batch size, N is the number of tokens, and D is the embedding dimension. The module consists of L layers, where each layer $i \in 1, 2, \ldots, L$ is defined by a weight matrix $\mathbf{W}_i \in \mathbb{R}^{Din_i \times Dout_i}$ and a bias vector $\mathbf{b}_i \in \mathbb{R}^{Dout_i}$. The forward pass of the STAMP module can be described as follows:

$$\mathbf{H}_1 = \text{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1) \tag{2}$$

$$\begin{gathered}\mathbf{H}_i = \text{ReLU}(\mathbf{H}_{i-1}\mathbf{W}_i + \mathbf{b}_i), \\ \forall i \in \{2, 3, \ldots, L-1\}\end{gathered} \tag{3}$$

$$\mathbf{M} = \mathbf{H}_{L-1}\mathbf{W}_L + \mathbf{b}_L \tag{4}$$

Where ReLU$(\cdot)$ represents the rectified linear unit activation function, $\mathbf{H}_i \in \mathbb{R}^{B\times N\times D_{\text{out}_i}}$ is the output of the i-th layer, and $\mathbf{M} \in \mathbb{R}^{B\times N\times N}$ is the generated mask. Note that the input $\mathbf{X}$ and all intermediate outputs $\mathbf{H}_i$ maintain the batch and number of tokens (B, N). For optimal implementation, STAMP's dimensionality should align with the Transformer Layer. In this single setup, the soft masking is computed once at the initial stage and subsequently propagated across all transformer layers.

3.3 Integration with Transformer Layers

We implement a soft token masking mechanism with context awareness for any type of Transformer Layer (Vaswani et al., 2017). For each Transformer Layer, with input $\mathbf{X}$, and learnable mask $\mathbf{M}_i$, the attention mechanism is expressed as:

$$\text{Attention} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} \diamond \mathbf{M}\right) \tag{5}$$

Here, Q and K are query and key projections of X, d_k is the key dimension, and $\diamond$ represents the fusion operation. This fusion can be implemented in two ways: addition ($A \diamond B = A + B$) or element-wise multiplication ($A \diamond B = A \odot B$). The element-wise multiplication, denoted by $\odot$, applies the operation to corresponding elements of the matrices. We tested both methods in our Ablation Studies (Section A.4.1). By default, we use element-wise multiplication in Section 4. When a different operation is employed, we specify it explicitly.

3.4 Multi-Layer Soft Token Attention Masking Fusion (Multi-Layer STAMP)

We extend STAMP approach to a stack of L Transformer layers, with each layer l having its own unique learnable mask, denoted as $\mathbf{M}_l$. The attention mechanism at each layer is computed according to Equation 5. The masks $\mathbf{M}_l$ are learned independently for each layer, allowing each STAMP to adapt its masking operation based on the layer's representation. As the output of one layer serves as the input to the next, this allows for hierarchical representation learning across the stack.

4 Experiments

4.1 Datasets

Generating AD. MADv2 (Soldan et al., 2022; Han et al., 2023c) is a vast dataset for video-language grounding, with over 264K queries in 488 movies totaling 892 hours. It includes MADv2-eval, with 10 movies for evaluation.

Moment Retrieval and Highlights Detection. QVHighlights (Lei et al., 2021) is the latest dataset for moment retrieval and highlight detection, featuring annotations for both tasks in over 10,000 YouTube videos.

4.2 Metrics

Generating AD. Conventional metrics like Rouge-L (R-L)(Lin, 2004), CIDEr (C)(Vedantam et al., 2014), and Retrieval-based metric (R@k/N) (Han et al., 2023b) are employed to compare generated Audio Descriptions (AD) with ground-truth AD. These metrics are robust to low-level variations in testing data, with higher values indicating superior text generation.

Moment Retrieval and Highlights Detection. For video grounding tasks, evaluation metrics include Recall@K and mAP@K for IoU=θ (R@K-

Model	R-L	C	R@5/16
LlaMA Adapter (Gao et al., 2023)	10.0(±0.65)	9.0(±0.35)	42.86(±0.55)
Ours	**13.54**(±**0.5**)	**18.56**(±**0.2**)	**56.15**(±**0.30**)
Gain(Δ)	3.54	9.56	13.29

(a) AD Task on MADv2-named (Soldan et al., 2022; Han et al., 2023c)

Model	R1@IoU0.7	mAP
QD-DETR (Moon et al., 2023)	44.98(±0.8)	39.86(±0.6)
Ours	**46.94**(±**0.6**)	**42.32**(±**0.7**)
Gain(Δ)	1.96	2.46

(b) Moment Retrieval Task in QVHighlights (Lei et al., 2021)

Model	mAP	HIT@1
QD-DETR (Moon et al., 2023)	38.94(±0.4)	62.40(±1.4)
Ours	**39.70**(±**1.0**)	**63.33**(±**0.8**)
Gain(Δ)	0.76	0.93

(c) Highlights Detection at VeryGood confidence in QVHighlights (Lei et al., 2021)

Table 1: **Comprehensive Performance Comparison across Tasks and Datasets.** This table reports the evaluation of our Multi-Layer STAMP-enhanced approach against established baselines across three tasks. On the AD task (MADv2-named dataset), our method significantly outperforms the LlaMA Adapter, achieving gains of 3.54 in R-L, 9.56 in C, and 13.29 in R@5/16. In the Moment Retrieval task on QVHighlights, our model surpasses QD-DETR with improvements of 1.96 in R1@IoU0.7 and 2.46 in mAP, while for Highlights Detection, it shows gains of 0.76 in mAP and 0.93 in HIT@1. These results underscore the robustness and effectiveness of our approach across diverse evaluation metrics and scenarios.

IoU=θ), assessing both ranking and temporal overlap. Models are evaluated at $K = 1$ with IoU thresholds of 0.5 and 0.7. Average mAP across IoU thresholds from 0.5 to 0.95 with 0.05 increments is calculated. Highlight detection primarily employs mAP, while HIT@1 measures the hit ratio for the highest scored clip.

4.3 Baselines

The proposed STAMP module can be incorporated into any of the existing Transformer Encoders. We integrated our contribution into two baseline models: LlaMA AdapterV2 (Gao et al., 2023) with a transformer-based audiovisual encoder, QD-DETR (Moon et al., 2023).

4.4 Results

Table 1 highlights that integrating Multi-Layer STAMP into existing models consistently improves performance across various tasks. In the AD Task on the MADv2-named dataset, our method achieves substantial gains over the LlaMA Adapter in Table 1a, with improvements of 3.54 in R-L, 9.56 in C, and 13.29 in R@5/16, indicating a robust enhancement in anomaly detection capabilities. Similarly, on the Moment Retrieval Task in QVHighlights (Table 1b), our model outperforms QD-DETR by 1.96 and 2.46 points in R1@IoU0.7 and mAP, respectively, and in the Highlights Detection task (Table 1c), it shows modest yet consistent gains of 0.76 in mAP and 0.93 in HIT@1.

Takeaway: Our method effectively harnesses multimodal sequences, showing significant gains in multimodal settings and highlighting its potential to advance multimodal learning. These findings indicate promising directions for further optimization and research across a range of task domains.

4.5 Ablation Studies

This section presents ablation studies evaluating the effectiveness of our STAMP module across multiple dimensions, including its impact on model performance, computational efficiency, and scalability. We explore the influence of the attention mask, compare STAMP with parameter scaling, and assess computational trade-offs. Additionally, we examine its applicability to unimodal sequences, investigate the role of Flash Attention (Dao, 2023) and dataset scaling, and identify potential limitations. Our analysis highlights that STAMP provides performance gains with minimal computational overhead.

Impact of Attention Mask Architectures. We conducted a comprehensive analysis of the performance impact of various attention mask architectures, focusing on our novel STAMP module. Evaluations were conducted on a specific subset

Mask	R-L	C
Full Attention	12.92	15.46
Sparse Learnable Mask*	10.02	9.72
STAMP	13.10	16.58
Multi-Layer STAMP	**14.28**	**17.11**

Table 2: **Attention Mask Influence.** We evaluate the performance of the Audio Description generation task using a subset of 1,010 instances from the MADv2 dataset (see Appendix A). Our results demonstrate that integrating the STAMP module enhances performance. Notably, the improvement is more pronounced in a multi-layer setup, where each transformer attention layer employs its own STAMP rather than relying on a global one, as seen in row 3. The '*' denotes the use of a learnable sparse mask design, following Lin et al. (2022). All experiments were conducted with identical hyperparameters and trained for 10 epochs.

Experiment	R-L	C	Param.
Baseline	12.92	15.46	6.93 B
Base. + Linear Layers	11.23	12.87	7.07 B
Base. + Transf. Layers	13.80	16.22	7.20 B
Multi-Layer STAMP	**14.28**	**17.11**	7.07 B

Table 3: **Parameter Count vs. STAMP Module Influence.** This table examines the impact of various architectural modifications on performance on the AD generation task, emphasizing models with comparable parameter counts. Our findings reveal that merely increasing the number of parameters by stacking linear layers (Row 2) does not necessarily enhance performance. A similar trend is observed when adding transformer layers (Row 3), indicating that parameter growth alone is not a sufficient factor for improvement. Notably, the Multi-Layer STAMP achieves the highest scores despite having fewer parameters than some configurations, underscoring its efficiency and effectiveness.

of the MADv2 dataset[1], with performance quantified using the Rouge-L and CIDEr metrics. As in Table 2, our investigation encompassed four distinct configurations: (i) full attention as a baseline, (ii) the learnable sparse mask from SwinBert, (iii) our proposed STAMP, and (iv) an extended Multi-Layer STAMP. Results demonstrate that the introduction of STAMP yields a substantial performance improvement over the full attention baseline, with CIDEr scores increasing from 15.46 to 16.58. Notably, the Multi-Layer STAMP architecture achieved superior performance, reaching a CIDEr score of 17.11. In contrast, SwinBERT's sparse learnable mask exhibited a marked decrease in performance, with CIDEr dropping from 15.46 to 9.72. We attribute this decline to the mask's inability to effectively capture the dynamic nature of MAD-v2 sequences, which are characterized by frequent shot changes, transitions, and complex audio-visual interactions. Our STAMP architecture shows improved performance in handling these multimodal sequences, indicating its effectiveness for the MAD-v2 video captioning task.

Parameters vs. STAMP Influence. To determine whether the performance gains of our STAMP module come from its unique design rather than simply having more parameters, we compared various baseline models by matching STAMP's parameter count. As shown in Table 3 (Row 2), simply increasing the number of parameters by adding additional linear layers at the end of each transformer stage unexpectedly led to a performance drop, reducing the CIDEr score from 15.46 to 12.87. This suggests that the additional parameters led to overfitting rather than improved learning. Next, we enhanced the audiovisual encoder in the baseline by adding more Transformer encoder layers (Row 3), increasing the total parameter count to about 7.2 billion—comparable to Multi-STAMP's 7.07 billion and slightly above the baseline's 6.93 billion. Although this upgraded baseline achieved higher scores than the original (Rouge-L: 13.80, CIDEr: 16.22), it still fell short of the Multi-Layer STAMP model's performance. These findings confirm that STAMP's advantage lies in its targeted token attention refinements, not just in having more parameters.

Computational Overhead. To address concerns about the additional computational complexity introduced by our module, we performed a detailed analysis of model efficiency. Building on our previous experiments (Table 3), we evaluated each model's computational overhead by measuring floating-point operations (FLOPs) and multiply-accumulate operations (MACs). Table 4 summarizes these metrics along with parameter counts and latency measurements for the baseline, baseline + linear layers, baseline + Transformer layers, and our proposed Multi-Layer STAMP models. Our analysis shows that the Multi-Layer STAMP architecture achieves a strong balance between performance and computational cost. Although it incurs a modest increase in FLOPs, MACs, and parameters compared to the baseline, it outperforms the baseline + liner layers experiment by achieving lower FLOPs and MACs, which translates to an intermediate latency profile. This result demonstrates that STAMP optimizes resource utilization efficiently while maintaining a balanced trade-off with speed. In contrast, incorporating additional Transformer

[1] See Appendix A.3.3

Model	Flops(↓)	MACs(↓)	Latency(↓)
Baseline	**3.39T**	**1.69T**	**87.57**
Base. + Linear Layers	3.45T	1.80T	89.12
Base. + Transf. Layers	3.58T	1.95T	94.33
Multi-Layer STAMP	3.43T	1.71T	88.60

Table 4: **Computational metrics for AD Generation in MADv2 (Soldan et al., 2022; Han et al., 2023c)** The table presents a comparison of models based on computational cost (FLOPs and MACs) and latency (milliseconds). The baseline model demonstrates moderate performance, whereas the Baseline + Linear Layers and Baseline + Transformer Layers experiment faces challenges due to increased resource consumption.

layers inherently increases computational complexity, primarily due to the attention mechanisms. This is evident in the rise in MACs from 1.69T to 1.95T, accompanied by a similar increase in latency and FLOPs.

Model	Acc-Top1(↑)	Acc-Top5(↑)
*ViT Base (He et al., 2021)	82.71	96.32
Ours	**83.45**	**96.59**
Gain(∆)	0.74	0.27

Table 5: **Image Classification on ImageNet 1K (Deng et al., 2009).** Performance comparison between ViT Base and our model, where higher values (↑) indicate better performance. "Ours" refers to ViT Base enhanced with the Multi-Layer STAMP module. The asterisk (*) denotes that we retrained using the codebase and observed a slight decrease in performance compared to the numbers reported in (He et al., 2021).

Single Modality. Although not the primary focus of this study, we conducted a supplementary exploration to assess the applicability of STAMP to unimodal sequences or a single modality, aiming to better understand its limitations. To this end, we evaluated two large datasets: ImageNet 1K (Deng et al., 2009) and MSRVTT (Xu et al., 2016). For ImageNet, we measured performance using top-1 accuracy, while for MSRVTT, we employed BLEU4 (B4) (Papineni et al., 2002), CIDEr (C), SPICE (S) (Anderson et al., 2016), METEOR (M) (Lavie and Agarwal, 2007), and Rouge-L (R-L) (Lin, 2004). The results, presented in Tables 5 and 6, show only marginal improvements. Specifically, the C and S metrics increased by 0.28 and 0.39, respectively, while Table 5 reports a modest gain of 0.74 in Acc-Top1. These findings indicate that STAMP is not particularly effective for single-modality encoders. The complexity inherent in multimodal data provides a richer context for STAMP to learn which tokens to prioritize. The increased diversity and variability across modalities (See Table 8) allow STAMP to refine its masking more effectively than in unimodal sequences, where such diversity is limited and the standard self-attention mechanism may already suffice.

Smooth Integration. To demonstate that our proposed module can be seamlessly integrated into various Transformer architectures, we conducted an experiment. In Table 7, we show the comparison between Multi-layer STAMP with and without Flash Attention V2 (Dao, 2023). The comparison reveals that while the RL (Rouge-L) and C (CIDEr) scores remain largely consistent, indicating minimal impact on output quality, substantial improvements are observed in computational efficiency. Specifically, FLOPs and MACs are significantly reduced from 3.43 TFLOPs to 2.272 TFLOPs and from 1.71 TMACs to 1.13 TMACs, respectively. Furthermore, latency is notably decreased from 88.60 ms to 64.30 ms, underscoring the enhanced processing speed. These results demonstrate that integrating Flash Attention V2 effectively optimizes computational performance while maintaining overall model effectiveness.

Dataset Scaling. Increasing the number of parameters significantly affects data scaling across modalities. Table 8 (extending Table 1) details performance gains on visual, text, and audio datasets of various sizes. We report each metric's maximum gain to emphasize major improvements and avoid averaging distortions. The results reveal a strong correlation between dataset complexity and STAMP's effectiveness. Notably, the large-scale multimodal MADv2 dataset (2.44 TB approx. from 488 films at 5GB approx. each) achieves the highest gains (13.29), especially with a multi-layer STAMP configuration. In contrast, smaller datasets like MSRVTT (6.3 GB) show modest gains (0.39), while medium-sized unimodal datasets such as ImageNet-1k (164 GB) see intermediate improvements (0.74). These findings underscore STAMP's capacity for handling complex, high-dimensional data and adapting to intricate cross-modal relationships, ultimately justifying its computational cost where traditional fine-tuning methods may fall short.

More Ablation Studies. Appendix A.4 examines the impact of depth on STAMP's performance, Appendix A.4.1 compares two element-wise operations for masking fusion, and Appendix A.4.2 analyzes how STAMP alters attention weight distri-

Model	**B4(↑)**	**R-L(↑)**	**M(↑)**	**C(↑)**	**S(↑)**
SwinBERT (Lin et al., 2022)	**42.82**	**62.06**	30.39	51.96	7.64
Ours	42.03	62.05	**30.60**	**52.24**	**8.03**
Gain(Δ)	−0.79	−0.01	0.21	0.28	0.39

Table 6: **Video Captioning Task on MSRVTT (Xu et al., 2016).** Evaluation of different models on the MSRVTT dataset. Higher values (↑) indicate better performance. "Ours" denotes the SwinBERT model with the Multi-Layer STAMP.

Attn	**R-L(↑)**	**C(↑)**	**F(↓)**	**M(↓)**	**P(↓)**	**L(↓)**
w/o FlashV2	**14.28**	17.11	3.43T	1.71T	7.07B	88.6
with FlashV2	14.19	**17.23**	**2.27T**	**1.13T**	7.07B	**64.3**

Table 7: **Computational and Performance Metrics for Multi-layer STAMP with and without Flash Attention V2.** This table compares standard Multi-layer STAMP to its Flash Attention V2 variant (Dao, 2023). While Rouge-L (R-L) and CIDEr (C) scores remain stable, computational efficiency improves significantly, reducing FLOPs, MACs, and latency from 88.60 ms to 64.30 ms. Lower values (↓) indicate better efficiency, while higher values (↑) reflect improved performance. (L) = Latency (ms), (C) = CIDEr, (R-L) = Rouge-L, (F) = FLOPs, (M) = MACs.

Dataset	**Size**	**Modality**	**Max. Gain**
MSRVTT	6.3 GB	V + T	0.39
Imagenet 1k	164 GB	V	0.74
QVHighlights	∼ 180 GB	V + T	2.46
MADv2	∼ 2.4 TB	V + A + T	13.29

Table 8: **Maximum Performance Gain by Dataset Size and Modality.** This table, adapted from Table 1, presents the maximum performance gains for each dataset, organized by size and modality (V: Vision, T: Text, A: Audio). The results show that larger datasets with multiple modalities, such as MADv2, typically achieve higher performance gains. Conversely, the smaller MSRVTT dataset demonstrates the lowest performance gain. The goal is to highlight the largest performance gaps, as averaging metrics can be misleading due to scale differences that may distort true disparities.

bution. We encourage the reader to explore these experiments for deeper insights.

Qualitative Analysis. We performed two qualitative analyses on the challenging Audio Description Task. The first two analyses focused on scenarios in which the video and audio streams are temporally aligned, examining how STAMP prioritizes specific tokens. A third analysis investigated instances where the streams are temporally misaligned. For further details, please refer to Appendix A.4.3.

5 Limitations

While STAMP excels in multimodal tasks, its effectiveness in unimodal settings is limited due to the lack of diverse cross-modal interactions that drive its adaptive token prioritization. In single-modality scenarios, self-attention already captures essential relationships, leaving less room for improvement through STAMP's masking mechanism. Additionally, the method introduces computational overhead, which, although optimized, may still pose challenges for real-time applications and resource-limited environments. Furthermore, STAMP relies on large-scale pre-trained models such as LLaMA 7B and CLIP, potentially inheriting biases and limiting its adaptability to smaller, task-specific models. Lastly, while it effectively refines attention, its interpretability remains an open area of research, requiring deeper analysis of how it selects and prioritizes tokens.

6 Conclusions

In this work, we introduced STAMP, a soft-masking mechanism designed to enhance multimodal learning by refining attention maps and dynamically prioritizing tokens. Our experiments on MADv2 and QVHighlights demonstrate that STAMP significantly improves performance in tasks such as audio description, video grounding, and highlight detection while maintaining low computational overhead. Ablation studies confirm that these improvements stem from adaptive token weighting without an increase in model parameters. Looking forward, future research should focus on further refining token weighting strategies and integrating STAMP with complementary attention methods to boost efficiency in real-time and resource-constrained settings. Additionally, advancing the interpretability of STAMP through visualization and explainability techniques could broaden its applicability across diverse, domain-specific models.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer.

Relja Arandjelovic and Andrew Zisserman. 2017. Look, listen and learn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 609–617.

Wayner Barrios, Mattia Soldan, Alberto Mario Ceballos-Arroyo, Fabian Caba Heilbron, and Bernard Ghanem. 2023. Localizing moments in long video via multimodal guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13667–13678.

C. Chen, Q. Fan, and R. Panda. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 347–356, Los Alamitos, CA, USA. IEEE Computer Society.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. 2019. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *Preprint*, arXiv:2307.08691.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Zhihao Fan, Yeyun Gong, Dayiheng Liu, Zhongyu Wei, Siyuan Wang, Jian Jiao, Nan Duan, Ruofei Zhang, and Xuanjing Huang. 2021. Mask attention networks: Rethinking and strengthen transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1692–1701, Online. Association for Computational Linguistics.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2018. Slowfast networks for video recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210.

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

Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, Xudong Lu, Shuai Ren, Yafei Wen, Xiaoxin Chen, Xiangyu Yue, Hongsheng Li, and Yu Qiao. 2023a. Imagebind-llm: Multi-modality instruction tuning. *Preprint*, arXiv:2309.03905.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. 2023b. AutoAD II: The Sequel - who, when, and what in movie audio description. In *ICCV*.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. 2023c. AutoAD: Movie description in context. In *CVPR*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. 2021. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. 2021. Mdetr - modulated detection for end-to-end multimodal understanding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1760–1770.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 228–231, USA. Association for Computational Linguistics.

Jun-Tae Lee, Mihir Jain, Hyoungwoo Park, and Sungrack Yun. 2021. Cross-attentional audio-visual fusion for weakly-supervised action localization. In

International Conference on Learning Representations.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, volume 34, pages 11846–11858. Curran Associates, Inc.

Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. 2021. Mst: Masked self-supervised transformer for visual representation. In *Advances in Neural Information Processing Systems*, volume 34, pages 13165–13176. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*.

Te Lin and Inwhee Joe. 2023. An adaptive masked attention mechanism to act on the local text in a global context for aspect-based sentiment analysis. *IEEE Access*, 11:43055–43066.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

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. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Riccardo Rende, Federica Gerace, Alessandro Laio, and Sebastian Goldt. 2024. What does self-attention learn from masked language modelling? *Preprint*, arXiv:2304.07235.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. 2022. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5026–5035.

Jingfan Tang, Xinqiang Wu, Min Zhang, Xiujie Zhang, and Ming Jiang. 2021. Multiway dynamic mask attention networks for natural language inference. *J. Comp. Methods in Sci. and Eng.*, 21(1):151–162.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

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

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. 2020. Multi-modality cross attention network for image and sentence matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10938–10947.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Mohit Bansal Yi-Lin Sung, Jaemin Cho. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *CVPR*.

Hang Zhang, Xin Li, and Lidong Bing. 2023a. Video-llama: An instruction-tuned audio-visual language model for video understanding. *Preprint*, arXiv:2306.02858.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Appendix

A.1 Multimodal Encoder Tasks

We evaluated the effectiveness of the Soft Token Attention Masking Process(STAMP) across three significant multimodal tasks: Audio Description (AD) Generation, Moment Retrieval, and Highlights Detection. In this section, we first outline the definitions of these tasks (Sections A.1.1 and A.1.2) and show the implementation details of our proposed STAMP module to each task (Sections A.1.3 and A.1.4).

A.1.1 Audio Description Generation

Our task involves adapting a Large Language Model (LLM) to generate Audio Descriptions (AD) in text for a long-form movie $\mathcal{L}$ segmented into short clips $\{c_1, c_2, \ldots, c_N\}$. Each clip encompasses $\mathcal{S}$ samples in the visual stream (represented as V) and S samples in the audio stream (denoted as A)[2]. Specifically, our goal is to create a text t_i that describes the audiovisual content presented in each clip c_i, aiming to assist individuals who are blind in following the movie's narrative.

Audiovisual Model $\mathcal{AV}$. We aim to train an audiovisual model that comprehends the relationships between sequential video and audio streams. Consequently, $\mathcal{AV}$ processes video (V) and audio (A) observations sampled at c_i clip and produces an audiovisual feature representations E_{va}.

$$\mathcal{AV}\,(V, A) \rightarrow E_{va} \tag{6}$$

Large Language Model $\mathcal{H}$. Given an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, the model $\mathcal{H}$ estimates the probability distribution of the next word x_{n+1} based on the context using the chain rule of probability:

$$P(x_{n+1}|X) = P(x_{n+1}|x_1, x_2, ..., x_n) \tag{7}$$

The model is trained by maximizing the likelihood of generating the correct sequence according to the training data. During inference, it predicts the most likely next word given the context. The model's weights θ are optimized through back-propagation and gradient descent to improve its language understanding and generation capabilities.

Adapter Module $\mathcal{P}$. Let's assume a pre-trained model with parameters represented by θ. The adapter layer introduces additional parameters for audiovisual understanding task, and these parameters can be denoted as ϕ. The output of the adapter layers can be represented as $P(x', \phi)$, where x' is the projected audiovisual features into the language space. So, the overall output of the model with the adapter layer can be written as:

$$\mathcal{F}(\mathcal{H}(x, \theta), \mathcal{P}(x', \phi)) \rightarrow t_i \tag{8}$$

Where $\mathcal{F}$ is a function that combines the pre-trained Language Model $\mathcal{H}$ and the adapter $\mathcal{P}$ to produce an AD in text t_i.

A.1.2 Moment Retrieval and Highlights detection

The visual-language grounding model, denoted as $\mathcal{G}$, is tasked with processing an untrimmed video, V, sampled from a temporal window W, along with a natural language query Q. It then produces predictions for J temporal moments, defined as:

$$\mathcal{G}\,(V, Q) \rightarrow (\tau_s, \tau_e, s, s_l)_1^J. \tag{9}$$

In Equation 9, the grounding models yield a series of moments ranked by their confidence scores. Here, (τ_s, τ_e) represents the duration span of the moment, while s indicates its confidence score. Now, let's define the inputs for our attention modules. Given a video comprising L clips and a text query containing N words, their representations extracted by frozen video and text encoders are denoted as $v_1, v_2, \ldots, v_L$ and $t_1, t_2, \ldots, t_N$, respectively. Additionally, the grounding model provides saliency scores s_l for each moment for the highlight detection task.

A.1.3 Implementation Details for AD Generation

Feature Extraction. The extraction of visual features follows the CLIP-based methodology outlined in (Soldan et al., 2022). To be more specific, visual features are extracted at a rate of 5 frames per second (FPS) with an embedding dimensionality of D_v=512. For audio feature extraction, we follow (Barrios et al., 2023) by utilizing the OpenL3 (Cramer et al., 2019; Arandjelovic and Zisserman, 2017) checkpoint pre-trained on videos containing environmental audiovisual data. We use a spectrogram time-frequency representation with 128 bands and set the audio embedding dimensionality D_a to 512. Furthermore, we extract the audio embeddings using a stride size of 0.2 seconds, *i.e.*,

[2] Raw sound from movies, excluding descriptions

with an extraction frame rate of 5 Hz, matching the frame rate of the visual features.

Audiovisual Model $\mathcal{AV}$. We utilize a Multimodal Transformer with a standard configuration (Vaswani et al., 2017). For each observation c_i, consisting of both visual and audio information, we employ $S = 25$ visual tokens and $S = 25$ audio tokens, effectively spanning a 5-second duration at a frame rate of 5 FPS. This Multimodal Transformer architecture comprises 16 layers and employs a Multi-Layer Soft Token Attention Masking ProcessModule with a dimensionality of 768 and depth of 16.

Large Language Model $\mathcal{H}$. For Large Language Model, we choose to employ a frozen LLaMA 7B model (Touvron et al., 2023) and opt to use its official checkpoint.

Adapter Module $\mathcal{P}$. We build our audiovisual adapter following the approach done in (Gao et al., 2023). In this part, we select 16 tokens as audiovisual tokens. We adjust the last 31 layers of LLaMA 7B, making sure that the audiovisual features stay at a size of 512, which then maps to 4096 (LLaMA dimensionality). We set the depth to 8, use 16 heads, apply LoRA Rank (Hu et al., 2021) with a value of 16, and activate Bias layers (Zhang et al., 2023b).

Training Protocol. To generate Audio Descriptions, we follow the training methodology outlined in (Zhang et al., 2023b; Gao et al., 2023). This involved utilizing 8 RTX 6000 Ada Generation GPUs, each equipped with 50 GB VRAM, alongside employing a base learning rate of $1e-4$ and the Adam optimizer.

A.1.4 Implementation Details for Moment Retrieval and Highlighting Task

Feature Extraction. The visual and text embeddings are extracted following the methodology presented in (Lei et al., 2021). For video, we use SlowFast (Feichtenhofer et al., 2018) and the visual encoder (ViT-B/32) of CLIP (Radford et al., 2021) to extract features every 2 seconds. We then normalize the two features and concatenate them at hidden dimension. The resulting visual features is denoted as $E_V \in \mathbb{R}^{L_V \times D_V}$, with $D_V = 2816$. For text features , we use the CLIP text encoder to extract token level features, $E_V \in \mathbb{R}^{L_Q \times D_Q}$ with $D_V = 512$.

Video Grounding Model. We adopt the methodology outlined in (Moon et al., 2023). The architecture consists of three distinct components: an encoder comprising four layers of transformer blocks (two cross-attention layers and two self-attention layers), while the decoder has only two layers. We configure the hidden dimension of the transformers to be 256 Additionally, for the transformer encoder layers and the cross-attention layers, we utilize our LAACM using dimensionality of 256 and depth of 32 layers.

Training Protocol. We conducted training over 200 epochs, employing a batch size of 32 and a learning rate set to $1e-4$. We utilized the Adam optimizer with a weight decay of $1e-4$, leveraging a single GPU, the RTX 6000 Ada Generation.

A.2 Single Modality Encoder Tasks

A.2.1 Image Classification Task

In the image classification task, the goal is to assign an input image I to one or more predefined classes from a set of C classes. Let's denote the image classification model as $\mathcal{M}$. Given an input image I, the model generates a set of class predictions and their corresponding confidence scores:

$$\mathcal{M}(I) \to (\hat{y}_1, \hat{p}_1), (\hat{y}_2, \hat{p}_2), \ldots, (\hat{y}_C, \hat{p}_C) \quad (10)$$

Here, $\hat{y}_c \in 1, 2, \ldots, C$ represents the predicted class label for the c-th class, and $\hat{p}_c \in [0, 1]$ is the corresponding confidence score or probability assigned by the model to that class. The model's goal is to accurately predict the true classes present in the input image I.

A.2.2 Video Captioning Task

In the video captioning task, the goal is to generate a textual description or caption for a given input video V. Let's denote the video captioning model as $\mathcal{M}$. Given an input video V, the model generates a sequence of words $W = w_1, w_2, \ldots, w_N$ that forms the caption:

$$\mathcal{M}(V) \to W = w_1, w_2, \ldots, w_N \quad (11)$$

Here, each w_i represents a word in the generated caption, and N is the length of the caption sequence. The model's objective is to produce a natural language caption W that accurately and coherently describes the content and events depicted in the input video V.

A.2.3 Implementations Details for Image Classification Task

We follow the pre-trained model developed in (He et al., 2021) and fine-tune it for the image classification task. The base model is a Vision Transformer

(ViT) with a 16x16 patch size, 768-dimensional embedding, 12 transformer layers, and 12 attention heads. It includes an MLP ratio of 4, biases in the query, key, and value projections, and layer normalization with an epsilon of $1e-6$. To incorporate our proposed Soft Token Attention Masking Process(STAMP) module, we use the Multi-Layer STAMP variant, which generates the attention mask using a single linear layer. For the pretraining stage, we adhere to the methodology outlined in (He et al., 2021), but increase the batch size to 128 and use 4 gradient accumulation steps. For fine-tuning on the image classification task, we maintain a batch size of 128 and 4 gradient accumulation steps. Additionally, we train for 100 epochs, apply a weight decay of 0.05, set the drop path rate to 0.1, and use mixup and cutmix with values of 0.8 and 1.0, respectively.

A.2.4 Implementations Details for Video Captioning Task

We adopt the methodology proposed by SwinBERT (Lin et al., 2022), with a notable modification. Instead of using a fixed learnable mask implemented via `nn.Parameter`, we integrate our Soft Token Attention Masking Process(STAMP) module, which consists of 16 layers while maintaining the same dimensionality as the original SwinBERT. Regarding the hyperparameters, the experiment utilizes a batch size of 2 per GPU, running for 20 epochs with a learning rate of 0.0003. Training is conducted in half precision using DeepSpeed, with gradient accumulation over 16 steps. For the entire training process, we used 8 A6000 Ada generation GPUs.

A.3 Additional Details for Audio Description Generation

In the following sections, we examine specific details that have not been addressed in the main paper. This comprehensive discussion includes insights into the current methodology for calculating metrics, the specific prompts employed, and the intricacies of both the training and evaluation processes for our implementation.

A.3.1 Metrics

In this work, we compute the CIDEr (Vedantam et al., 2014) score using the `pycocoeval` package from the coco-caption repository, adhering to the standard parameters of $n = 4$ and $sigma = 6$ as prescribed in (Vedantam et al., 2014). For Rouge-L (Lin, 2004), a commonly used metric in natural language processing, we leverage the Hugging Face `evaluate` library for implementation (evaluate-metric/rouge). The Rouge-L configuration is set with `use_aggregator=True` and `use_stemmer=True`, aligning with the default settings to ensure consistent evaluation. Prior to metric computation, both predicted and reference texts are normalized by converting to lowercase and removing punctuation, following standard preprocessing protocols.

For retrieval-based evaluation, we adopt the R@k/N metric, utilizing the methodology introduced in (Han et al., 2023b). This is further supplemented by the BERTScore (Zhang et al., 2020) metric, ensuring alignment with state-of-the-art retrieval practices. To maintain reproducibility and result comparability, we use the specified hash code for BERTScore: `roberta-large_L17_noidf_version=0.3.12(hug_trans=4.30.2)-rescaled`, which reflects the model version and Hugging Face environment at the time of evaluation. These standardized configurations and consistent preprocessing steps reinforce the robustness and reliability of our evaluation pipeline.

A.3.2 Natural Language Prompting

To implement Audio Description functionality in our model, we apply the prompting approach developed in the LLaMA Adapter framework (Gao et al., 2023). The primary prompt used for generating Audio Descriptions is: **"Below is an instruction that describes a task. Write a response that appropriately completes the request."** We then include a task-specific instruction: **"Generate a caption for this video."** This prompt setup, shown in Figure S3, provides the model with the necessary context to produce relevant and concise descriptions for the video content.

```
Below is an instruction that
    describes a task
### Instruction:
Generate caption of this
    video.
### Response:
```

Figure S3: **Prompt for Audio Description Generation** The caption provided outlines the prompt utilized to activate the functionality of Audio Description generation employing the LLaMA model.

A.3.3 Dataset Split

As MADv2 lacks a validation set, we curated a subset of 1010 moments from two movies, `3034_IDES_OF_MARCH` and `3074_THE_ROOMMATE` from the Unnamed version for our ablation studies and model selection. All models and experiments were assessed under consistent parameters to ensure fair comparisons. However, Table 1 in the main paper was generated using the entire dataset in the named version to maintain parity with other baselines.

A.3.4 Training Protocol

The training procedure for our Audio Description Generation model adhered closely to the methodology outlined in (Gao et al., 2023). The process began with an initial alignment phase aimed at ensuring robust synchronization between the audiovisual features. This phase was crucial for establishing coherence between the audio and visual modalities of the input data. Upon successful alignment, we resumed training with a focus on optimizing the bias and gate layers as proposed by (Gao et al., 2023), leveraging the LLaMA (Touvron et al., 2023) 7B architecture in combination with our audiovisual encoder. In this subsequent stage, we performed backpropagation exclusively on the bias, gate, and audiovisual layers to enhance the model's capacity to generate accurate and contextually relevant audio descriptions.

Training was conducted over a span of 20 epochs, with model selection based on performance on the validation subset. Hyperparameters were meticulously tuned, including a learning rate of $1e^{-4}$, weight decay of 0.05, and a batch size of 256. We employed the AdamW optimizer to ensure efficient parameter updates. During the audiovisual alignment phase, the adapter and audiovisual layers were trained for 2 epochs, with the rest of the model parameters held constant, facilitating stable convergence. Importantly, the LLaMA model's core parameters remained frozen throughout the entire training process, preserving the integrity of its pre-trained features while allowing focused adaptation of the newly introduced layers. This careful balance between alignment and fine-tuning was critical for achieving high-quality audio description generation without disrupting the foundational capabilities of the LLaMA architecture.

A.4 Ablation Studies

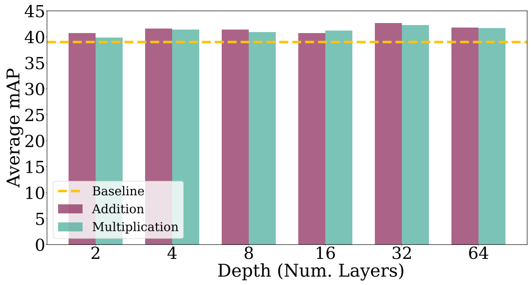


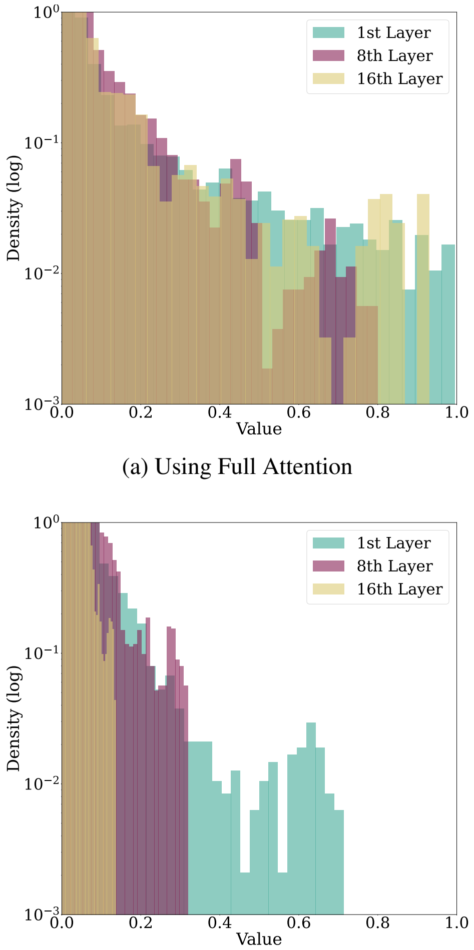
Figure S4: **Ablation Studies on the Number of Layers in STAMP and Types of Mask Operation.** We investigate the impact of varying the number of layers in the Soft Token Attention Masking Process (STAMP) module within a cross-attention configuration, as well as different methods for fusing the mask with attention weights. The experiments explore layer counts ranging from 2 to 64, and compare two distinct fusion techniques: element-wise multiplication and addition. Evaluation on the QVHighlights (Lei et al., 2021) validation set reveals notable improvements in the Average mAP metric for the Moment Retrieval task, with the most significant gains observed using 32 layers in the STAMP module.

A.4.1 Effects of Depth and Masking Fusion Techniques

Our ablation study on the cross-attention mechanism systematically investigates the impact of two critical components within the Soft Token Attention Masking Process (STAMP) module: the depth of the STAMP architecture and the mask operations (addition and multiplication). As illustrated in Figure S4, we evaluate performance using the Average mAP metric for Moment Retrieval on the QVHighlights validation set. The results demonstrate that STAMP consistently enhances performance across various configurations, with some architectures yielding more substantial improvements than others. Notably, all STAMP variants, regardless of layer composition or operation type, outperform the baseline model. The most effective configuration utilizes 32 layers with both addition and multiplication operations, achieving Average mAP scores of 42.61 and 42.32, respectively. These findings underscore the efficacy of our approach in bolstering Moment Retrieval performance on the QVHighlights dataset and suggest that the interplay between architectural depth and diverse mask operations is crucial for optimizing cross-attention mechanisms in this context.

A.4.2 Attention Weights

Figure S5 presents a comparative analysis of attention weight distributions across three critical layers (1st, 8th, and final) of the Transformer architec-



(a) Using Full Attention

(b) Using Multi-Layer STAMP

Figure S5: **Attention Weight Distribution.** This figure illustrates the effect of our STAMP on the distribution of attention weights during the AD generation task. When using STAMP, attention weights tend to decrease in magnitude as they propagate through deeper layers, with many approaching zero. This observation may facilitate future exploration of attention optimization by potentially reducing redundant computations. The attention weights shown were collected from a forward pass using 64 samples.

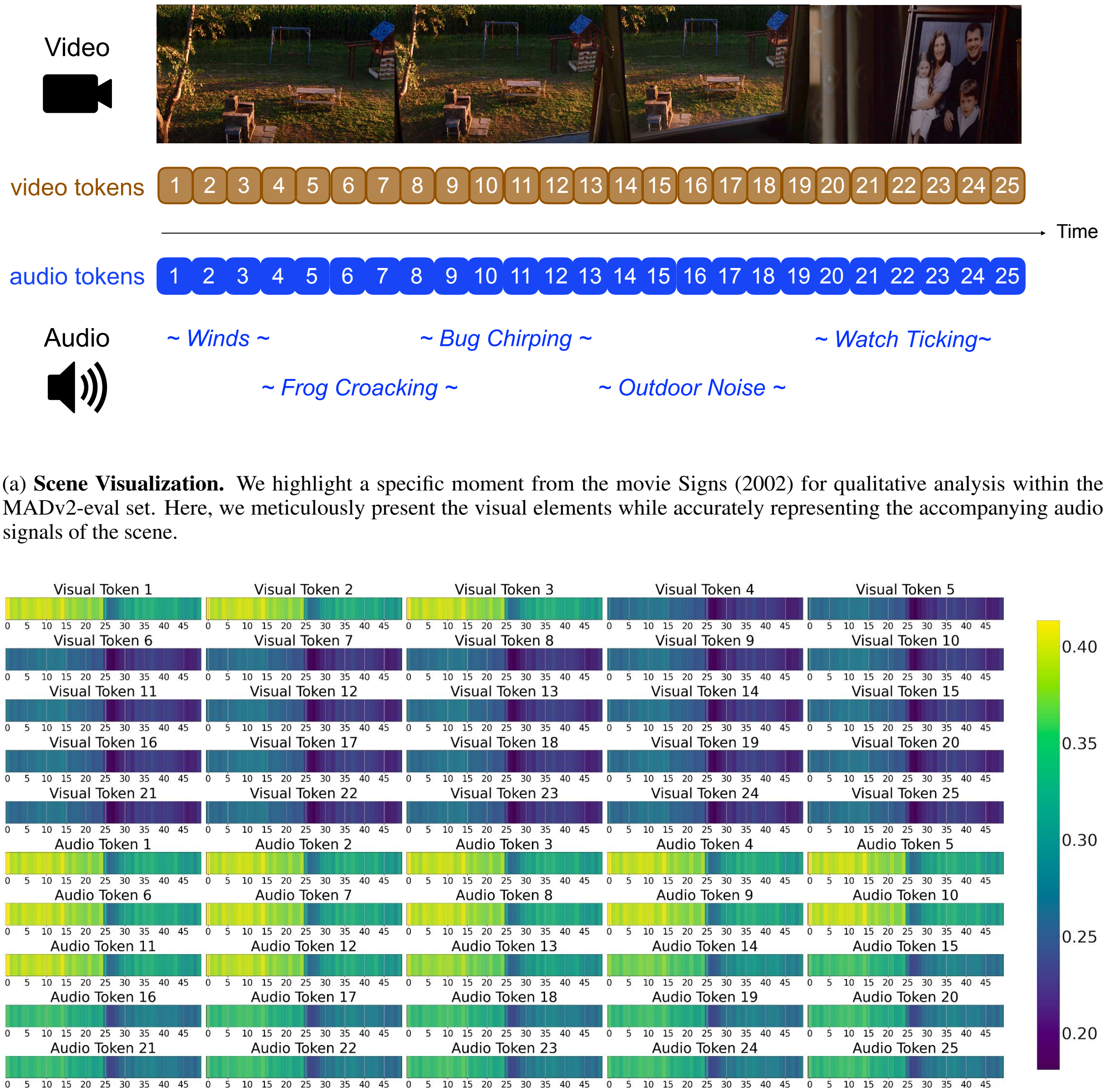
ture, contrasting traditional full-attention mechanisms with our proposed Multi-Layer Soft Token Attention Masking Process(STAMP). Our findings reveal a striking pattern: the implementation of Multi-Layer STAMP induces a substantial sparsification of attention weights, with a significant proportion reducing to zero and many others converging to near-zero values. This phenomenon suggests that STAMP effectively prunes redundant connections within the attention mechanism, potentially leading to more computationally efficient model training without sacrificing performance. The observed sparsity not only aligns with recent trends in neural network optimization but also opens avenues for further research into the interpretability and efficiency of attention-based models. While these results underscore the potential of STAMP as a promising approach for enhancing the scalability and resource utilization of Transformer-based architectures, there remains considerable room for improvement and further investigation. Future work could explore the optimal degree of sparsity, the impact on various downstream tasks, and potential hybridization with other attention optimization techniques to further push the boundaries of efficient, high-performance Transformer models.

In Table 7, we show the comparison between Multi-layer STAMP and Multi-layer STAMP with Flash Attention V2 (Dao, 2023). The comparison reveals that while the RL (Rouge-L) and C (CIDEr) scores remain largely consistent, indicating minimal impact on output quality, substantial improvements are observed in computational efficiency. Specifically, FLOPs and MACs are significantly reduced from 3.43 TFLOPs to 2.272 TFLOPs and from 1.71 TMACs to 1.13 TMACs, respectively. Furthermore, latency is notably decreased from 88.60 ms to 64.30 ms, underscoring the enhanced processing speed. These results demonstrate that integrating Flash Attention V2 effectively optimizes computational performance while maintaining overall model effectiveness.

A.4.3 Qualitative Analysis

Figure S6 presents a qualitative analysis of our Multi Layer Soft Token Attention Masking Process(STAMP) implementation for the Audio Description generation task. This visualization encompasses two key aspects: Figure S6a displays the concurrent audio and video signals, and Figure S6b illustrates the mask values corresponding to each token in the initial transformer layer. In this figure, the x-axis represents the sequence of tokens, and the colored heatmap indicates the mask values for each token in relation to the other tokens in the sequence. In this example, the first 25 tokens represent visual information, and the last 25 tokens correspond to the audio data. Each token (highlighted in the title) is analyzed in terms of its interaction with the sequence.

In this example, the token sequence has a shape of $(1, 50, 756)$, where 50 denotes the total number of tokens resulting from the concatenation of visual and audio tokens, each contributing 25 tokens. The visual content remains largely static across frames, depicting a residential backyard with minor visual variations. The auditory content transitions



(a) **Scene Visualization.** We highlight a specific moment from the movie Signs (2002) for qualitative analysis within the MADv2-eval set. Here, we meticulously present the visual elements while accurately representing the accompanying audio signals of the scene.

(b) **Scene Visualization** We also showcase the mask values produced by the Soft Token Attention Masking Process (STAMP) module for each visual and audio token present in the scene. These mask values exhibit positive numerical values, ranging between 0 and 1 inclusively.

Figure S6: **Qualitative Analysis.** This illustration presents a qualitative analysis of a specific instance from the MADv2-eval dataset. It depicts visual and audio signals alongside mask values corresponding to the initial transformer layer (1st layer). Video tokens are represented on the x-axis from 0 to 24, while audio tokens range from 25 to 49 on the same axis. The ground truth label for this moment is: "A set of swings and a climbing frame stand in a rural backyard, along with a picnic table and a brick barbecue."

from ambient sounds such as wind, insects, and outdoor noise to the rhythmic pattern of a clock. The ground truth Audio Description states: "A set of swings and a climbing frame stand in a rural backyard, along with a picnic table and a brick barbecue."

In this scenario, the STAMP module activates only three out of twenty-five visual tokens while assigning minimal attention to audio tokens. Figure S6b shows the masking values for each token, where the x-axis corresponds to the sequence of tokens resulting from the concatenation: tokens numbered from 0 to 24 are visual tokens, and tokens from 25 to 49 are audio tokens. For instance, for visual token 1, STAMP assigns values greater than 0.35 to tokens 0 to 24 (the visual tokens), in-

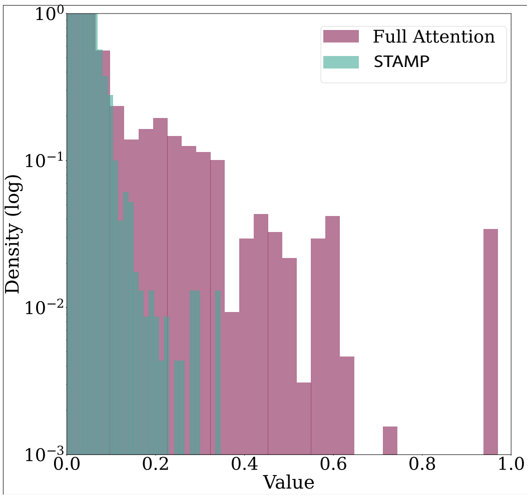


Figure S7: **Analysis of Attention Weight Distribution in the Qualitative Example.** The plot illustrates the distribution of attention weights within the initial transformer layer across two distinct configurations: employing STAMP and full-attention mechanisms. It is evident from the depiction that attention weights under STAMP influence tend to exhibit a leftward bias, resulting in a significant portion approaching 0 or nearing zero. The distribution weights correspond to the same example in Figure S6.

dicating strong correlations between these visual elements, as shown by the yellow-green-colored cells in the heatmap. Moreover, some correlation with audio tokens (25 to 49) is also visible in the figure, though these values are generally lower. Conversely, for audio token 1, STAMP assigns higher values to the initial visual tokens and lower values to the later visual tokens, reflecting the static nature of the visual information—a backyard scene with minimal dynamic changes—while the other audio tokens receive varying degrees of attention. Notably, the last audio tokens (e.g Audio Token 15 to 25) correspond to indoor sounds, indicating a scene transition from an outdoor to an indoor setting. Consequently, STAMP assigns values less than 0.35 in its masking for these tokens, interpreting them as less important and less related to the predominantly outdoor visual and audio tokens.

To compare with self attention, Figure S7 shows the attention weight distributions for both STAMP and full attention on the same scene. Without STAMP, the distribution is more uniform, suggesting that attention is spread across more tokens. With STAMP, the distribution is skewed to the left with many weights near zero, implying focused attention on fewer, more relevant tokens. This analysis highlights STAMP's capability to discern and prioritize specific tokens, thereby enhancing multimodal scene interpretation.

Figure S8 illustrates a challenging scenario for our STAMP approach. This example uses the same visual content as in Figure S6 but pairs it with audio samples comprising 25 tokens from the credits section, containing only background soundtrack music. While STAMP correctly assigns minimal values to the visual tokens, recognizing the lack of relevance between the video and the new audio tokens, it struggles to handle the audio tokens optimally. Instead of assigning values close to zero to the audio tokens—as would be expected given the irrelevance of the soundtrack to the scene description task—STAMP assigns intermediate values from its distribution. This outcome suggests potential areas for improvement in the model's audio-visual integration capabilities, particularly in distinguishing between relevant and irrelevant audio information.

Another example, depicted in Figure S9, involves the ground truth labeled as "They stop when they reach a gap." The scene opens with an image of a maize field, accompanied by the sudden sound of a little girl screaming. The film's protagonists immediately begin sprinting through the field, generating a distinct crunching noise alongside the sound of rapid footsteps. While the visual content is highly dynamic—both the character and the environment are in motion—the scene remains largely focused on the maize field and the actors running through it. This continues until a moment of silence marks their exit from the field. In Figure S9b, the final visual tokens (24 and 25) carry the most weight in the STAMP output because they show the characters stopping, which aligns with the ground truth. Additionally, the audio of the crunching sound (audio token 1 to 14) from the maize field provides context, as it reflects the running action and comes before the stopping action, which is the task's consequence. The silence that follows signifies the stopping action and the fact that the gap has been crossed, as there is no more maize field to traverse. This is why later audio tokens (20-25) are attended to, though less strongly, as they represent the conclusion of the scene.

In summary, these findings highlight both the strengths and limitations of STAMP in multimodal Audio Description generation. While the model effectively prioritizes relevant tokens in scenes with aligned audio and visual content, it struggles with irrelevant audio, assigning undue attention to non-informative tokens. This underscores the need for further refinement in its ability to discriminate between pertinent and extraneous information, sug-

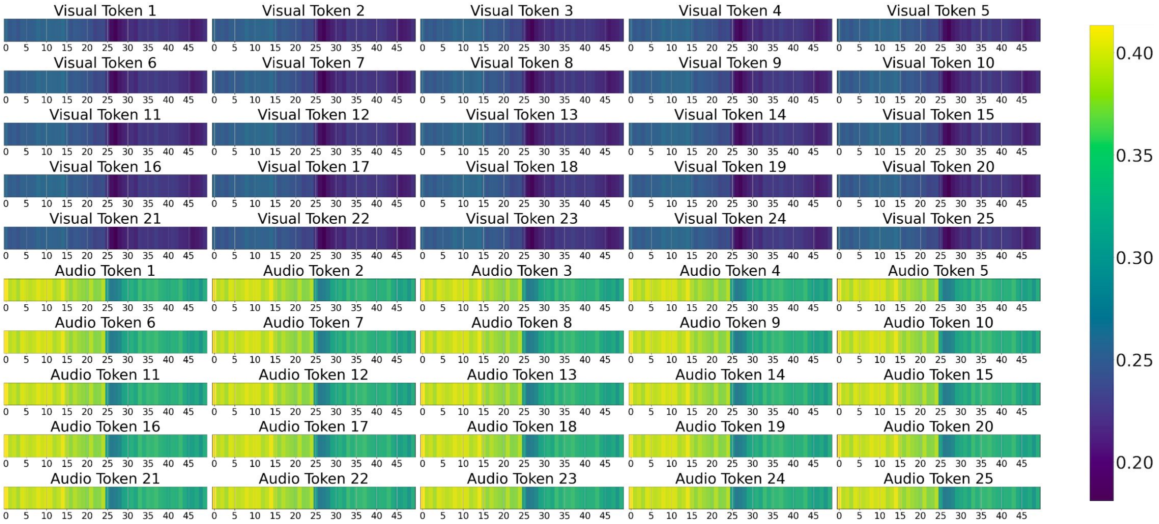
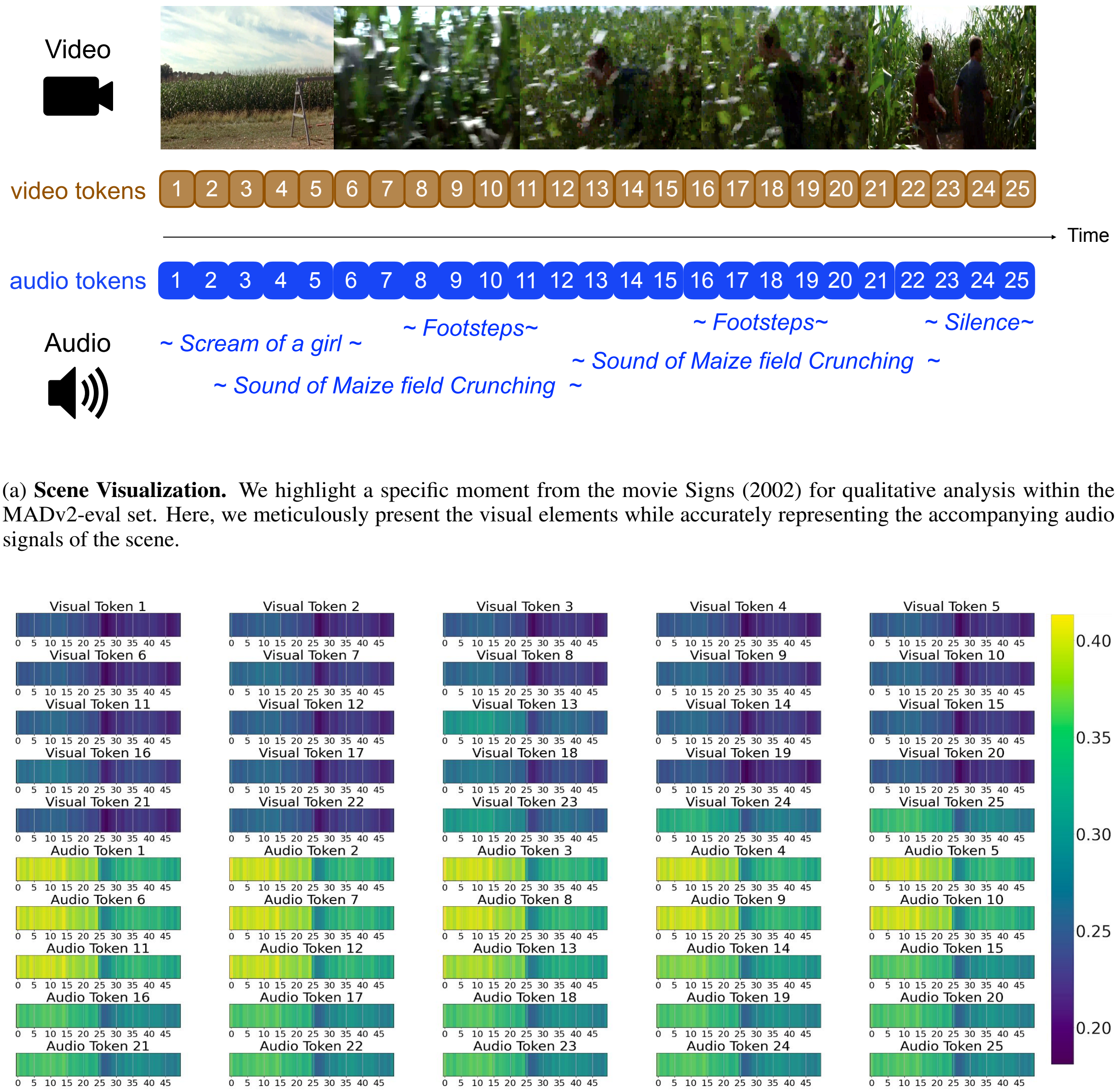


Figure S8: **Analysis of STAMP Failure Example in Audio Description Generation.** This plot illustrates the learned mask (STAMP's output) from the example shown in Figure S6. In this scenario, the visual features remain unchanged, but the audio tokens correspond to the last 25 samples from the movie's credits, which consist solely of the soundtrack. While the mask correctly assigns low values to the visual features, it fails to do so for the audio features, assigning mid-range values from the distribution instead. The x-axis represents the video tokens (ranging from 0 to 24) and the audio tokens (ranging from 25 to 49) on the same axis.

gesting avenues for future research to enhance multimodal attention mechanisms. There is still room for improvement, and we are optimistic that addressing these challenges will further advance the effectiveness of STAMP in complex multimodal tasks.



(a) **Scene Visualization.** We highlight a specific moment from the movie Signs (2002) for qualitative analysis within the MADv2-eval set. Here, we meticulously present the visual elements while accurately representing the accompanying audio signals of the scene.

(b) **Scene Visualization** We also showcase the mask values produced by the Soft Token Attention Masking Process (STAMP) module for each visual and audio token present in the scene. These mask values exhibit positive numerical values, ranging between 0 and 1 inclusively.

Figure S9: **Additional Example for Qualitative Analysis.** This illustration provides an additional example of qualitative analysis from the MADv2-eval dataset. It displays both visual and audio signals along with corresponding mask values from the first transformer layer (1st layer). The x-axis represents video tokens from 0 to 24 and audio tokens from 25 to 49. The ground truth label for this moment is: “They stop when they reach a gap".