Hierarchical Pre-Training of Vision Encoders with Large Language Models

Supplementary Material

6. Hierarchical vs. Cascaded Pre-Training

In this section, we compare hierarchical and cascaded pre-training approaches for vision encoders, highlighting their impact on gradient propagation and feature integration.

Hierarchical Pre-Training Hierarchical pre-training establishes direct cross-attention between multiple layers of the vision encoder and the LLM. By enabling gradient flow across different levels of abstraction, this approach preserves fine-grained information while allowing deeper layers to refine high-level representations. As a result, hierarchical pre-training fosters better feature propagation and improves overall model convergence.

Cascaded Pre-Training In contrast, cascaded pre-training follows a sequential learning process where only the final layer of the vision encoder interacts with the LLM. This approach reduces computational complexity but leads to attenuated gradient signals in earlier layers. Consequently, lower-layer features are less effectively incorporated into the model, potentially limiting performance in tasks requiring detailed visual understanding.

Key Observations As illustrated in Figure 4, hierarchical pre-training offers superior gradient flow, leading to better optimization and improved feature learning. The cascaded approach, while computationally more efficient, may hinder the ability to leverage low- and mid-level vision features effectively.

Our experimental results (detailed in Section 4) demonstrate that hierarchical pre-training significantly enhances classification accuracy, particularly for vision-language tasks requiring fine-grained feature alignment. These findings suggest that hierarchical cross-attention is a more effective strategy for integrating vision encoders with LLMs.

7. Hyperparameters

Pre-Training We outline the optimization hyperparameters and data augmentations used during HIVE pre-training in Table 3. For tokenization, we adopt the tokenizer used by SigLIP [44] and truncate any text longer than 77 tokens.

Classifier Fine-Tuning The optimization hyperparameters used during classifier fine-tuning are detailed in Table 4. For all experiments, we train a lightweight classifier on top of the frozen pre-trained vision encoder to evaluate the learned visual representations. This ensures a fair comparison across different pre-training approaches.

VLM Fine-Tuning for Vision-Language Tasks For vision-language model evaluation, we adopt a two-stage fine-tuning process based on the LLaVA [41] framework:

- **Stage 1: Connector Training.** We train the connector module to align the vision encoder's output with the LLM token space while keeping both the vision encoder and LLM frozen.
- **Stage 2: LLM Fine-Tuning.** We freeze the vision encoder and train the LLM on downstream vision-language datasets to refine the model's reasoning capabilities.

The hyperparameters used in both stages are detailed in Table 5.

8. Computational Efficiency

We evaluate the computational efficiency of hierarchical cross-attention compared to self-attention methods. Table 6 presents the measured training cost and memory overhead during model pretraining.

Our method achieves improved efficiency by applying cross-attention to only **25%** of the vision encoder layers, significantly reducing the number of attended tokens. Despite the reduced computational cost, our model consistently outperforms self-attention-based models across both classification and vision-language tasks (see Section 4.2).

These results highlight the efficiency advantages of our method, which achieves improved performance despite lower computational cost during training.

8.1. Computational Complexity Analysis

In this section, we analyze the computational complexity of hierarchical cross-attention in comparison to full self-attention, particularly in the context of vision-language models (VLMs). Self-attention mechanisms are widely used in vision-language pretraining, but they introduce significant computational overhead when processing high-dimensional visual inputs. The proposed hierarchical cross-attention mechanism offers a more efficient alternative by selectively integrating multi-level vision features into the large language model (LLM), reducing redundant computations.

Self-Attention Complexity In standard vision-language models, self-attention is applied across the full set of vision and text tokens. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, the vision encoder tokenizes it into N_v visual tokens, where

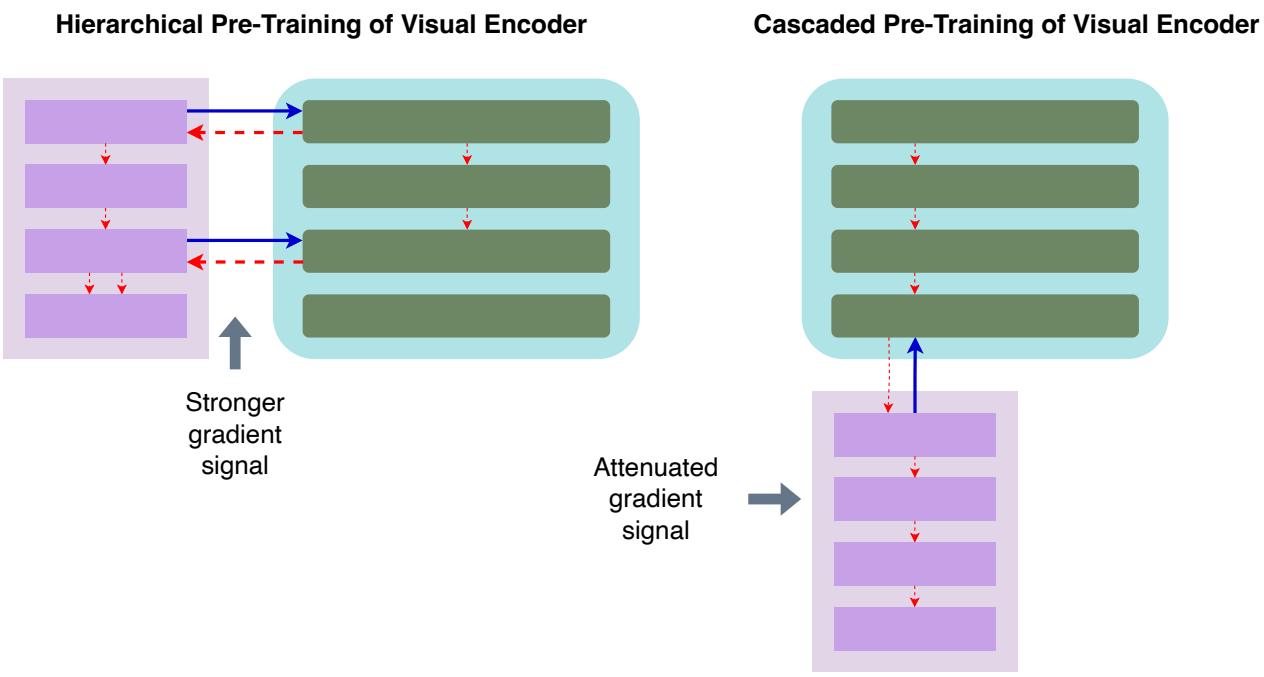

Figure 4. Comparison of hierarchical and cascaded pre-training approaches for vision encoders. Hierarchical pre-training (left) establishes direct cross-attention across multiple layers, allowing for stronger gradient propagation and better feature integration. Cascaded pre-training (right) restricts interactions to the final layer, leading to attenuated gradient signals and weaker hierarchical feature learning.

Table 3. Pre-training hyperparameters for the three-stage pre-training procedure used in HIVE.

Stage	Stage 1 (Projector)	Stage 2 (Projector + LLM)	Stage 3 (Full Model)
Optimizer	Fully decoupled AdamW [25]		
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.95$	$\beta_1 = 0.9, \beta_2 = 0.95$
Peak learning rate	1×10^{-3}	2×10^{-5}	2×10^{-6}
Minimum learning rate	1×10^{-4}	2×10^{-6}	0
Weight decay	0	0	0
Batch size	256	256	1024
Epoch	1	2	1
Gradient clipping	1.0	10.0	10.0
Warmup iterations	70	140	18
Total iterations	2326	4652	581
Learning rate schedule	Cosine decay [25]		

$N_v = HW/P^2$ and P is the patch size. The total number of tokens is given by:

$$N = N_v + N_t$$

where N_t is the number of text tokens.

The complexity of full self-attention in the LLM is then:

$$\mathcal{O}\left(L_l \frac{N^2 d}{2} + L_l N d^2\right) \tag{4}$$

where: - L_l is the number of transformer layers in the LLM, - $N^2/2$ arises from the causal masking in self-attention, which prevents future tokens from being attended to, - d is the hidden dimension, - The second term, $\mathcal{O}(L_l N d^2)$, corresponds to the MLP complexity for processing all tokens in the transformer layers.

For high-resolution images, this quadratic term N^2 and extensive MLP computation become significant bottlenecks, making full self-attention costly for large-scale vision-language pretraining.

Hierarchical Cross-Attention Complexity The proposed hierarchical cross-attention mechanism reduces com-

Table 4. Classifier fine-tuning hyperparameters for HIVE and baselines.

Config	Setting
Optimizer	AdamW [25]
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
Peak learning rate grid	2×10^{-4}
Minimum learning rate	0
Weight decay	0.01
Batch size	512
Gradient clipping	3.0
Warmup epochs	1.5
Learning rate schedule	Cosine decay
Augmentations	
RandomResizedCrop	
Scale	[0.4, 1.0]
Ratio	[0.75, 1.33]
Interpolation	Bicubic
RandomHorizontalFlip	$p = 0.5$
ColorJitter	
Brightness	0.2
Contrast	0.2
Saturation	0.2
Hue	0

Table 5. Hyperparameters for VLM fine-tuning on vision-language tasks.

Config	Stage 1 (Connector Training)	Stage 2 (LLM Fine-Tuning)
Optimizer	AdamW [25]	AdamW [25]
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$
Peak learning rate	1×10^{-3}	2×10^{-5}
Minimum learning rate	0	0
Weight decay	0	0
Batch size	256	64
Gradient clipping	1.0	1.0
Warmup iterations	66	347
Total iterations	2180	11540
Learning rate schedule	Cosine decay	Cosine decay

Table 6. Computational efficiency comparison between self-attention and HIVE. Values are reported relative to the self-attention baseline.

Method	**Relative Efficiency**
Training Cost (TFLOPs)	
Self-Attention	1.0×
HIVE (Ours)	0.14×
Memory Overhead	
Self-Attention	1.0×
HIVE (Ours)	0.8×

putational cost by restricting interactions to a subset of vision encoder layers and selectively integrating multi-level features into the LLM. Instead of processing all N_v visual tokens at every layer, hierarchical cross-attention operates on a subset $\mathcal{S}$ of informative layers from the vision encoder, each containing reduced feature representations.

Since visual tokens are not processed by the LLM's MLP layers, this design introduces substantial savings.

The complexity of hierarchical cross-attention is:

$$\mathcal{O}\left(L_l L_s d^2 + L_l N_t d^2\right) \tag{5}$$

where: - L_s is the number of selected vision encoder layers contributing to cross-attention, - N_t is the number of text tokens that still pass through the LLM's MLP layers.

Since $L_s \ll N_v$, hierarchical cross-attention avoids

the quadratic explosion of visual tokens seen in full self-attention and eliminates redundant MLP computations.

Comparison and Trade-offs Hierarchical cross-attention significantly reduces computational overhead compared to full self-attention while maintaining strong performance across downstream visual tasks. The complexity comparison is summarized in Table 7.

Table 7. Computational complexity comparison between full self-attention and hierarchical cross-attention in vision-language models.

Method	**Complexity**
Self-Attention (Causal)	$\mathcal{O}\left(L_l \frac{N^2 d}{2} + L_l N d^2\right)$
Cross-Attention (Causal)	$\mathcal{O}(L_l L_s d^2 + L_l N_t d^2)$

Full self-attention provides maximum feature interactions but becomes impractical for high-resolution vision-language tasks due to its quadratic scaling with the number of visual tokens and extensive MLP computations. This results in high computational costs and memory overhead, limiting scalability.

In contrast, hierarchical cross-attention selectively integrates multi-level visual features into the LLM, reducing redundant computations and significantly lowering computational requirements. By bypassing the LLM's MLP layers for visual tokens, hierarchical cross-attention achieves substantial savings while maintaining strong performance in both visual and multimodal tasks. Experimental results demonstrate that cross-attention achieves superior performance across fine-grained and large-scale classification benchmarks, reinforcing its effectiveness in downstream visual tasks.

8.2. Runtime and Efficiency Analysis

We empirically evaluate the computational efficiency of our hierarchical cross-attention framework compared to a full self-attention baseline pretrained using the LLaVA 1B model. While theoretical complexity is discussed in Section 8.1, here we measure actual runtime and memory usage to demonstrate practical scalability.

Experimental Setup We measure the following metrics averaged over five runs:

- *Wall-clock training time*: Duration per training epoch.
- *Memory consumption*: Peak GPU memory usage during training.

All evaluations use identical training parameters and batch sizes, ensuring fair comparisons.

Training Time Comparison Table 8 summarizes the average training epoch duration. Hierarchical cross-attention significantly reduces training time by limiting interactions to selected encoder layers.

Table 8. Average wall-clock training time per epoch. Hierarchical cross-attention provides notable speedups over self-attention.

Model	Training Time (min/epoch)	Speedup
Self-Attention	240	-
Cross-Attention	70	3.43×

Memory Consumption Peak GPU memory consumption during training is presented in Table 9. Our method considerably reduces memory requirements by decreasing token-level computations.

Table 9. Peak GPU memory usage during training. Hierarchical cross-attention reduces memory overhead substantially.

Model	Peak Memory (GB)	Reduction
Self-Attention	54.2	-
Cross-Attention	22.03	59.3%

9. Gradient and Attention Map Visualizations

In this section, we present qualitative visualizations of gradient flow and attention maps to illustrate the impact of hierarchical cross-attention on feature extraction and alignment.

9.1. Gradient Flow Analysis

Figures 5 to 11 present gradient map visualizations for various sample images. Each figure shows the original input (leftmost image), followed by gradient maps visualized across successive layers from the first to the final layer of the vision encoder.

Observations. HIVE produces sharper and more structured gradient distributions compared to self-attention models. In earlier layers, gradients are highly granular, effectively capturing fine visual details such as textures, edges, and object boundaries. As the network deepens, the gradients become progressively broader, focusing on higher-level semantic regions. This behavior reflects the hierarchical nature of HIVE's cross-attention mechanism, where low-level features are preserved while higher-level layers capture more abstract concepts.

The improved gradient flow is particularly evident in complex scenes, where key visual elements such as objects, text, and motion cues are effectively emphasized. This

structured gradient propagation contributes to HIVE's enhanced stability during training and improved visual representation learning.

Conclusion. These visualizations highlight HIVE's ability to promote stable gradient flow by efficiently distributing gradients across encoder layers. The observed improvements in feature refinement and gradient stability align with HIVE's enhanced performance across vision-language and classification benchmarks.

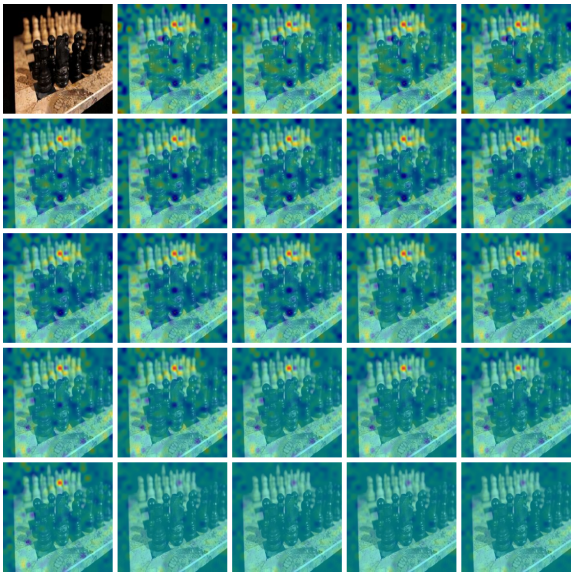

Figure 5. Gradient map visualization for Sample 1: "Investigators and journalists gather around the car of person after an attack on Wednesday." Gradients emphasize individuals and vehicles with sharp, localized activations in early layers.

9.2. Attention Map Analysis

Figures 12, 13, and 14 present visualizations of attention maps produced by HIVE's cross-attention layers. Compared to self-attention methods, HIVE's cross-attention achieves sharper and more meaningful activations, improving token-to-region alignment.

In Figure 12, corresponding to the caption "Investigators and journalists gather around the car of person after an attack on Wednesday," HIVE effectively highlights key elements such as the car and surrounding individuals. The focused attention on these subjects illustrates HIVE's ability to capture crucial semantic details in complex environments.

Figure 13, corresponding to the caption "Colorful plastic and aluminum chairs leaning against tables at a cafe outdoor dining area," shows HIVE's ability to isolate distinct objects, particularly the chairs and tables. The focused activations align with the scene's core visual features, emphasizing HIVE's improved object localization.

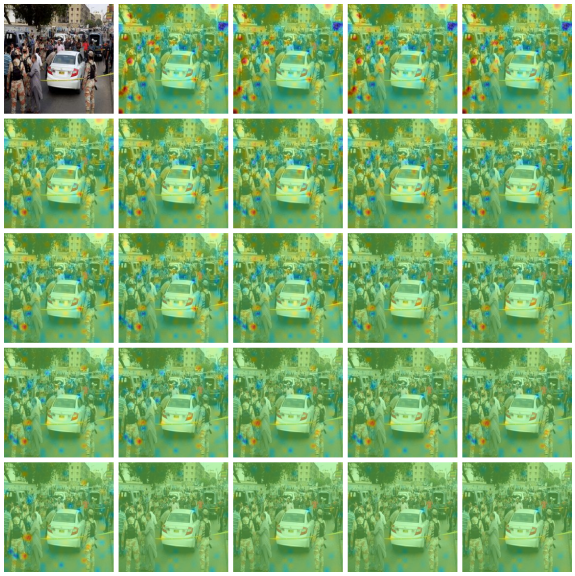

Figure 6. Gradient map visualization for Sample 2: "Colorful plastic and aluminum chairs leaning against tables at a cafe outdoor dining area." Early layers highlight fine-grained details such as chair edges, while deeper layers emphasize broader scene structure.

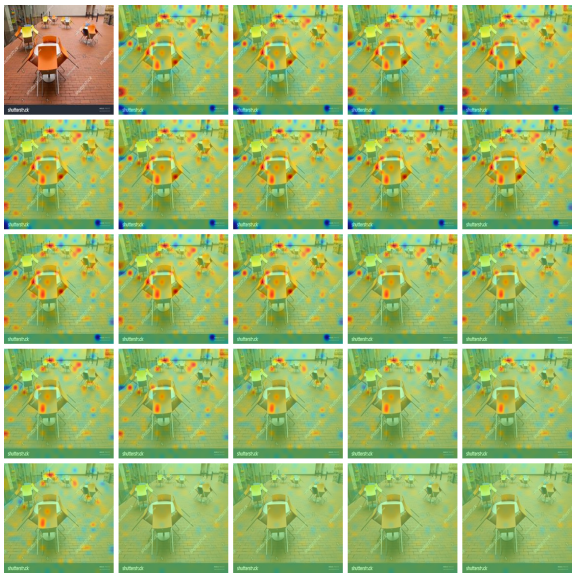

Figure 7. Gradient map visualization for Sample 3: "Person makes a move on defenders during the spring game." HIVE captures dynamic motion cues, focusing on the athlete and defenders.

In Figure 14, corresponding to the caption "Person makes a move on defenders during the spring game," HIVE effectively emphasizes the athlete's movement and surrounding players. This behavior highlights HIVE's strength

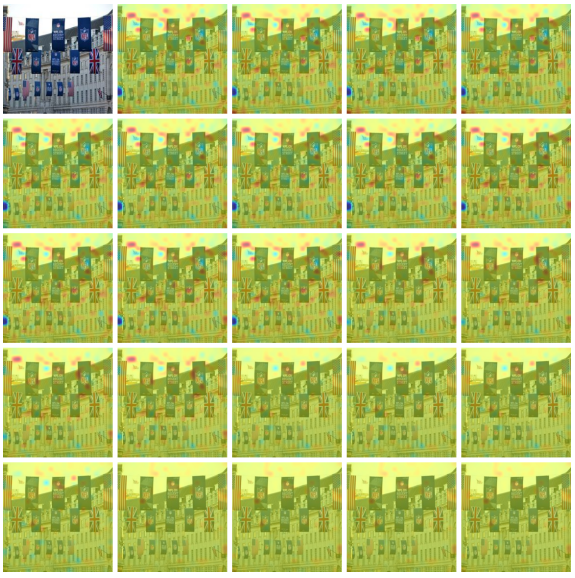

Figure 8. Gradient map visualization for Sample 4. HIVE maintains granular focus on key visual features, improving gradient flow.

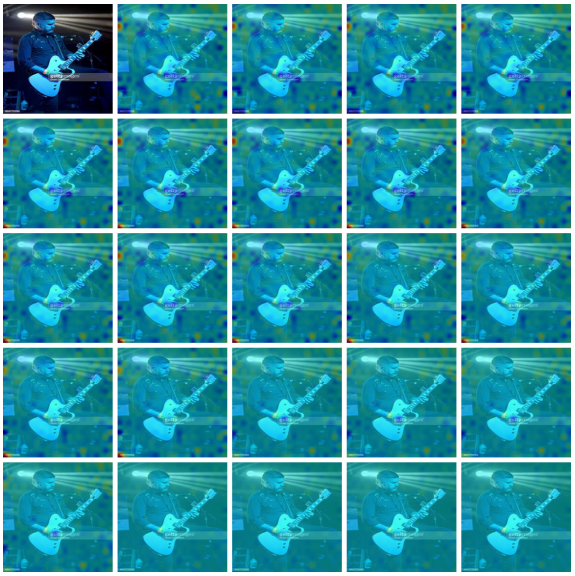

Figure 9. Gradient map visualization for Sample 5. Enhanced gradient stability enables sharper feature refinement in early layers.

in capturing dynamic visual cues and distinguishing key elements in action-driven scenarios.

Conclusion. These visualizations illustrate that HIVE's cross-attention mechanism effectively integrates both low-level and high-level visual features. By dynamically attending to task-relevant regions, HIVE enhances visual grounding, improving performance across complex visual scenes

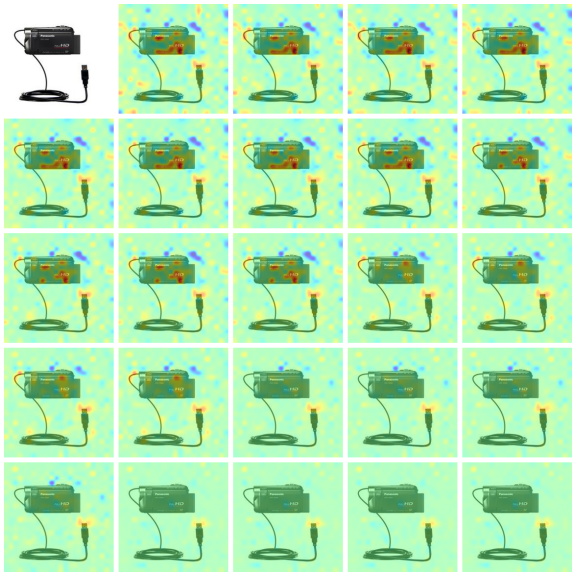

Figure 10. Gradient map visualization for Sample 6. HIVE consistently emphasizes meaningful visual elements across encoder layers.

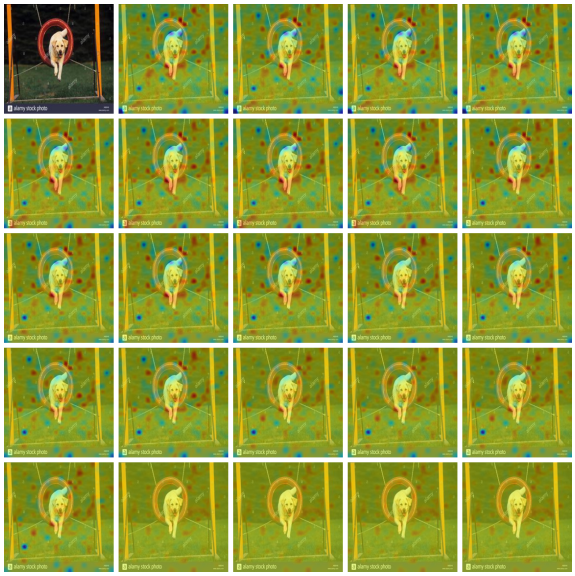

Figure 11. Gradient map visualization for Sample 7. Stable gradient propagation ensures effective visual feature learning across hierarchical layers.

in both static and dynamic environments.

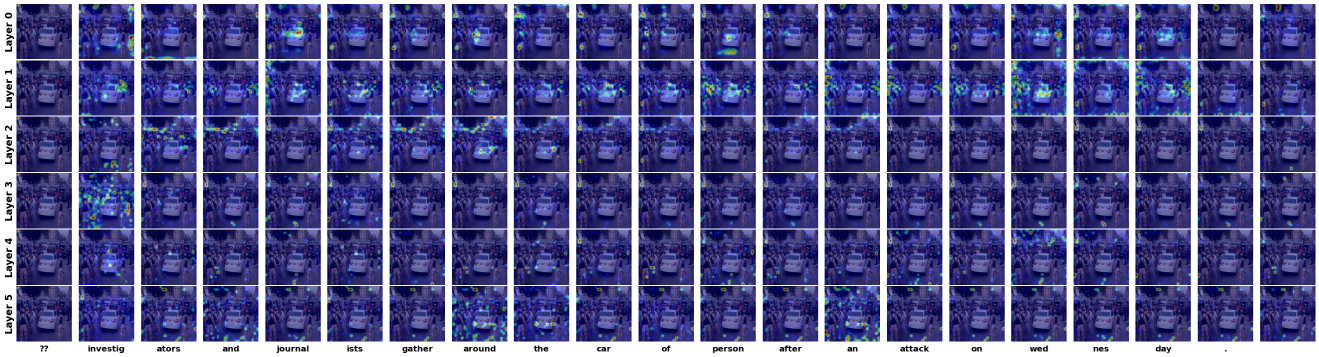

Figure 12. Attention map visualization for Sample 1: ”Investigators and journalists gather around the car of person after an attack on Wednesday.” HIVE emphasizes key elements such as the car and surrounding individuals, demonstrating improved semantic localization.

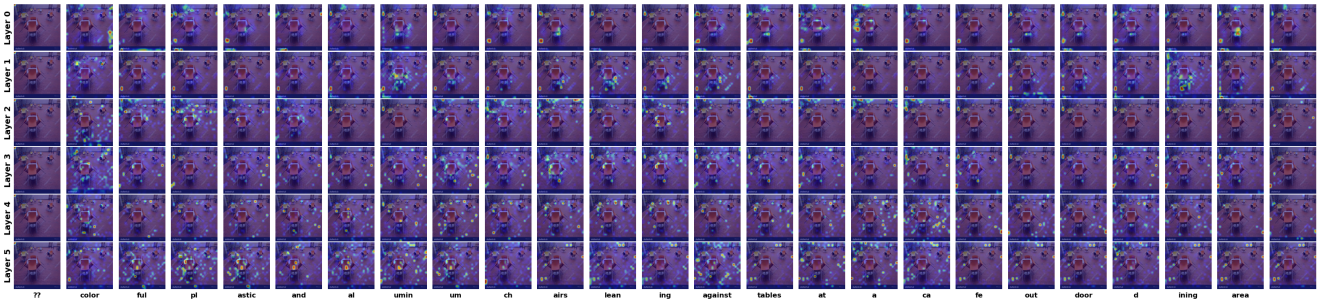

Figure 13. Attention map visualization for Sample 2: ”Colorful plastic and aluminum chairs leaning against tables at a cafe outdoor dining area.” HIVE highlights chairs and tables with sharper focus, enhancing object-level localization.

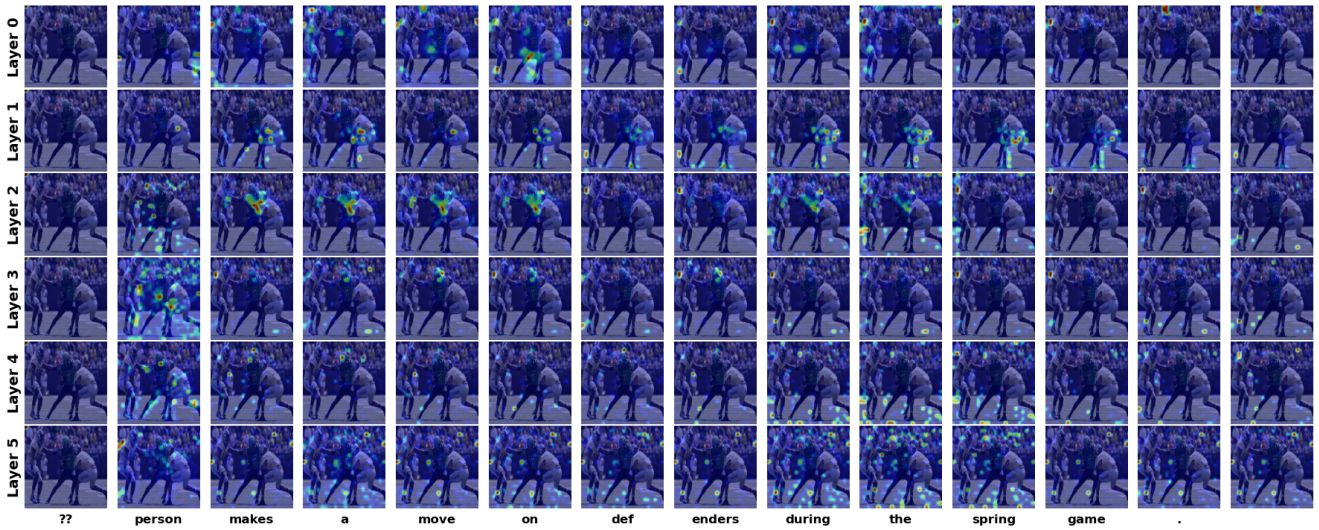

Figure 14. Attention map visualization for Sample 3: ”Person makes a move on defenders during the spring game.” HIVE effectively highlights the athlete’s movement and surrounding players, improving focus on dynamic elements.