

6 Appendix

In this appendix, we present supplementary material related to our study. This includes related work, additional information on the dataset creation process and the design of the prompts used in our experiments. We also include results from some ablation studies conducted during our research. Finally, we provide additional details about the model training process to assist readers who may wish to reproduce our work from scratch.

Short table of Notations To ensure that the readers can directly refer to a table for certain notations without scouring through the text, we provide a table of notations for quick reference where the symbols are accompanied by a short description.

Table 6: Summary of key notations used in temporal gaze aggregation and occlusion filtering.

Symbol	Description
t	Timestamp of current frame
δ	Temporal aggregation window size (e.g., 200ms)
I_t	RGB image frame at time t
g_t	Gaze point (pixel coordinate) at time t
m_t	Spatial heatmap generated from g_t via Gaussian smoothing
H_t	Aggregated gaze heatmap at time t after occlusion filtering
$f_{\tau \rightarrow t}$	Forward optical flow from frame I_τ to I_t
$f_{t \rightarrow \tau}$	Backward optical flow from I_t to I_τ
\mathbf{p}	Designated pixel location in image I_τ
$\hat{\mathbf{p}}$	Translated pixel location in I_t using forward flow
Δ	Discrepancy between forward and backward flow vectors
η_{observed}	Proportion of pixels with flow discrepancy $> \epsilon$
ϵ	Threshold for pixel-level flow discrepancy
η	Threshold for deciding major occlusion (e.g., 60%)
o_τ	Binary occlusion validity flag for frame I_τ

6.1 More Related Work

Vision Language Models: VLMs take in as input images and text together ($image, text$) and produce a text output. VLMs learn a mapping function ($image, text \rightarrow text$) where the task varies from Visual Question Answering (VQA) tasks to generating text based on image and text provided. The VLMs output text condition on the image text sequence and several models exist such as BLIP, LLaVa, Flamingo etc. [Liu et al., (2023)], [Li et al., (2022)], [Zhang et al., (2023)], [Alayrac et al., (2022)], [Chen et al., (2023)]. In this study, we initially employ the open-source version of the Flamingo model [Awadalla et al., (2023)], built upon the foundations of the original Flamingo described in [Alayrac et al., (2022)]. To show that our approach can be implemented and generalized to other architectures, we also utilise LaViLa’s Narrator module [Zhao et al., (2022)], and adapted versions of InternVL [Chen et al., (2024)] and OpenLLaVA [Lin and Long, (2024)] in our experiments. These models were chosen for their use of attention mechanisms, which are central to our method—our framework explicitly aims to modulate attention to better align with human visual focus.

Dataset The Ego4D and EPIC-Kitchens datasets consist of egocentric videos of camera-wearers performing daily activities in semi-controlled environments [Grauman et al., (2022); Damen et al., (2022)]. Additional relevant datasets include the EGTEA+ Gaze dataset, with 28 hours of cooking-centric content paired with gaze [Li et al., (2020b)], and the Visual Data Experience (VDE) dataset, which provides approximately 240 hours of everyday activity recordings with synchronized gaze and head tracking [Greene et al., (2024)].

While some datasets provide coarse labels, the gaze-augmented subset of Ego4D used in our work lacks fine-grained annotations. To address this, we supplement it with descriptive captions generated via GPT-4V. Future work may extend this setup by incorporating VDE or other multimodal datasets.

Additionally, [Huang et al. \(2025\)](#) introduce a dataset combining egocentric and exocentric perspectives for procedural activities. This dual view setup which aligns with an "observe first, imitate next" paradigm may enhance downstream modeling but the inclusion is left for future work.

Attention and gaze-augmented models: Attention-based models are widely used to identify important features and improve performance across various domains, including autonomous driving ([Braunagel et al., 2017](#)), action prediction, and human-computer interaction ([Weber et al., 2020](#); [Shaffi et al., 2019](#); [Aronson et al., 2018](#)). These models enhance interpretability by directing focus to the most relevant regions of an image or sequence. For instance, class activation maps have been employed to leverage pooling layers in deep networks, generating class saliency maps ([Sudhakaran and Lanz, 2018](#)) that highlight key objects associated with future actions. Guided-attention mechanisms have further been used to model next-active object interactions, improving action anticipation ([Thakur et al., 2023](#)). The Spatiotemporal Attention Module (STAM) ([Lu et al., 2019b](#)) incorporates eye gaze as supervision, training a network to predict attention maps for activity recognition. Other studies have similarly used gaze to guide attention maps in activity recognition tasks ([Min and Corso, 2020](#); [Awale and Sarikaya, 2022](#)). Beyond artificial systems, human perception and action have long been studied in relation to gaze. Eye movements provide task-specific visual cues, enabling more efficient action execution ([Hayhoe et al., 2003](#)). In our work, we explore how eye gaze data can be integrated into Vision-Language Models (VLMs) to improve egocentric behavior understanding, conditioning text annotations on gaze data to enhance future activity prediction. In our project, we study various ways to utilize eye gaze data in a VLM setting, building a gaze-regularized attention mechanism. Our model conditions predictions on eye gaze data as an input signal to output fine-grained text annotations of events happening in the near future, as well as providing detailed annotations of the current activity.

6.2 Dataset Creation

In this section, we provide details about the dataset creation process used for model training and testing as well as how the occlusion filtering mechanism is utilized to create aggregated spatial heatmaps.

6.2.1 Dataset Construction and Prompt Design

The Ego4D dataset comprises egocentric video clips along with supplementary data such as audio, text annotations, eye gaze data, and additional metadata ([Grauman et al., 2022](#)). For our project, we focused on the subset of video clips that include eye gaze data, containing approximately 33.3 hours of egocentric videos recorded from 80 participants. The eye gaze data is provided in numerical form, containing canonical timestamps and the pixel coordinates of gaze points. We transform gaze points to images to represent important visual regions, aligning with how humans perceive spatial information ([Laeng et al., 2014](#)). Due to the minute differences between consecutive images in the original videos, we perform downsampling to one image per second to reduce computational requirements while remaining effective. Since our focus is fine-grained egocentric behavior understanding, we modify existing data with detailed textual descriptions to enhance human-machine interaction. We leverage GPT-4V to generate annotations for video frames by processing a sequence of images and prompting it to describe each frame. This method ensures contextual coherence across frames. After obtaining initial descriptions, we evaluate their quality and provide feedback to refine the output. This feedback is used to modify the prompt, which is then fed back into GPT-4V with the image frames. We conduct prompt selection and refinement using multiple sequences from different video clips to ensure generalizability. This iterative process continues until we establish an optimal prompt template that consistently yields accurate and contextually appropriate annotations.

Specifically, we start by passing a small set of images (validation set) to GPT-4V with a basic prompt like: "Describe what is happening in the image sequence and output the text descriptions." The initial output was manually evaluated, and feedback was provided. This feedback, along with the original prompt, was passed to ChatGPT to refine the prompt for generating accurate and meaningful text annotations. The manual evaluation process ensured that the generated annotations met the following criteria:

- A clear description of the objects being manipulated or focused on in the scene.

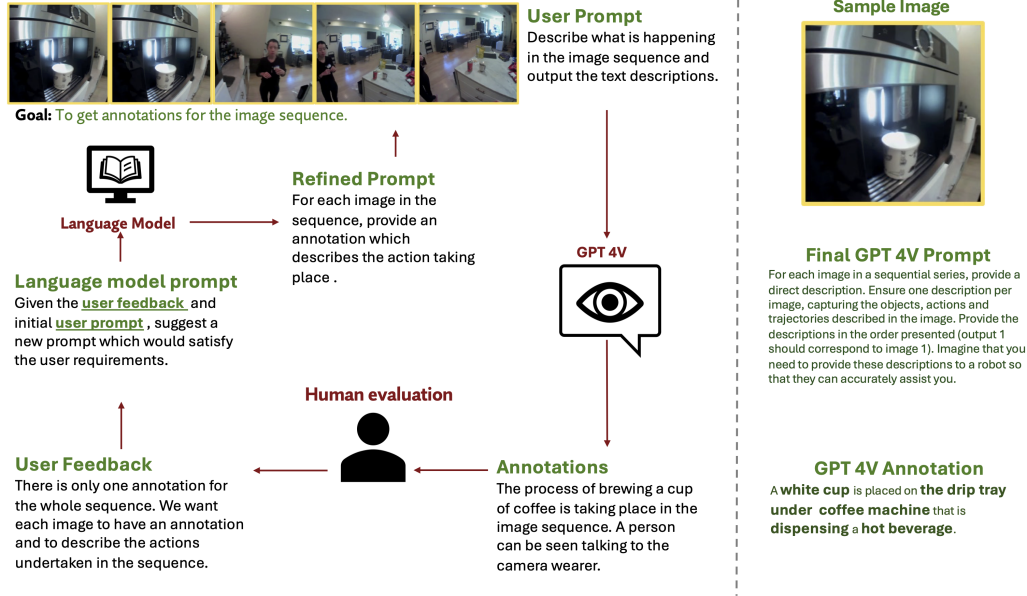


Figure 5: **GPT-4V Annotation Workflow Left.** We illustrate our annotation pipeline, where sequences of egocentric images are provided to GPT-4V along with an iteratively refined prompt. Human feedback is used to improve the prompt over multiple rounds, ensuring greater accuracy and contextual coherence in the generated captions. The use of the entire sequence helps maintain temporal consistency, allowing GPT-4V to better capture object interactions and ongoing actions. *Right.* We show a sample image from the dataset, the final refined prompt, and the resulting annotation. This process results in finer-grained text annotations which we utilise in our experiments.

- A detailed account of the actions being performed by both the camera wearer and other individuals present in the images.
- Information about any trajectories or movements that take place within the scene.
- Clear, fine-grained annotations that fulfill these criteria in a way that is easily understandable by both humans and machines, such as robots that may use these instructions for task execution.

After several iterations of refining the prompt and evaluating the results, we identified a suitable template for generating high-quality text annotations. This final template was then used to annotate the image sequences using GPT-4V. More details on the prompt design process can be found in Figure 5 and for a corresponding high level view on the algorithm please refer to Algorithm 1.

We also provide additional details regarding the dataset used in our work. To reiterate, the purpose of using a third-party captioning process was to enhance the granularity of the usual action describing annotations i.e make it more fine-grained. We had approximately 33 hours of data and since the images are sampled at every 1 second, we had a little over 108000 images. The images were divided into 'chunks' each of about size 10 such that when they are provided to GPT-4V, GPT-4V is aware of the full context. Due to connection issues, sometimes we would get an error and so for such cases, we would discard the chunk. In addition, certain annotations also contained the word 'blurred' or 'unclear', which is not useful for egocentric behaviour understanding tasks and we tried to avoid such annotations-image pair as much as possible. Due to the above two reasons, we ended up losing around 5500 images. Annotations typically ranged from 15–20 words and terms like 'individual' and 'camera-wearer' were common. Their overuse might have contributed to some annotation inaccuracies. Two human evaluators were responsible for iteratively refining the GPT-4V prompts and verifying the quality of generated annotations. They evaluated five sets of ten images from ten chosen clips (i.e., 50 images per set, 500 images total) to identify where the annotations were inaccurate or incomplete. Priority was given to identify and flag instances where the annotations are completely wrong and irrelevant. This iterative prompt-refinement and evaluation process took approximately 1.5 weeks in total including the time required to set up the automated code.

Algorithm 1 Iterative Prompt Refinement for Fine-Grained Annotation

Input: Validation image subset $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_m\}$;

Full image set $\mathcal{I} = \{\mathcal{I}_1, \dots, \mathcal{I}_k\}$;

Initial prompt P_0 ;

GPT-4V model; ChatGPT interface

Output: Final refined prompt P^* ;

Annotations $\{\mathcal{T}_1, \dots, \mathcal{T}_k\}$

1 $P \leftarrow P_0$

2 **repeat**

3 Generate annotations $\hat{\mathcal{T}}_{\mathcal{V}} \leftarrow \text{GPT4V}(\mathcal{V}, P)$

4 Manually evaluate $\hat{\mathcal{T}}_{\mathcal{V}}$ for:

- Object relevance and accuracy
- Actions of camera wearer and others
- Movements or spatial cues
- Clarity and fine granularity

Provide feedback F on $\hat{\mathcal{T}}_{\mathcal{V}}$

Refine prompt: $P \leftarrow \text{ChatGPT}(P, F)$

5 **until** annotations on \mathcal{V} are satisfactory;

6 $P^* \leftarrow P$; // Final prompt after refinement on validation set

7 **for** each batch \mathcal{I}_j ($j = 1$ to k) **do**

8 Generate final annotations:

$\mathcal{T}_j \leftarrow \text{GPT4V}(\mathcal{I}_j, P^*)$

9 **return** $P^*, \{\mathcal{T}_1, \dots, \mathcal{T}_k\}$

6.3 Occlusion-Check Process

The input image sequence can exhibit dynamic changes due to movement in the environment or the movement of the camera wearer. The method for aggregating gaze points is suitable only when the frames within the δ interval for which aggregation is done, shows moderate movement. However, preventing movement in a dynamic environment is challenging. If we aggregate gaze points in the $[t - \delta, t]$ interval to construct the aggregated heatmap H_t and there is major occlusion between the earlier frames $[t - \delta, t)$ and the final frame at timestamp t , it becomes impractical to include gaze points from the occluded frames. In such cases, collecting gaze points from occluded frames may lead to inaccurate or misleading representations in the heatmap. To alleviate this issue, we perform an occlusion check between each frame in the $[t - \delta, t)$ interval and the final frame at time t to ensure appropriate gaze aggregation. In the case of significant occlusion or a drastic change in the scene, gaze points corresponding to the earlier frames should not be collected for the aggregation and subsequent formation of the heatmap H_t .

Using a method similar to the consistency check with optical flow presented by [Hur and Roth \(2017\)](#), we explicitly exclude gaze points that are occluded in the current frame. If a pixel is correctly translated and there is no major occlusion, then the difference between the forward optical flow displacement of pixel \mathbf{p} and the displacement of the translated pixel $\hat{\mathbf{p}}$ with backward optical flow should be close to zero.

For an RGB image I_t at time t , we gather the image frames $\{I_\tau\}$ for all $\tau \in [t - \delta, t)$. Let the forward optical flow between images I_τ and I_t in the horizontal direction be denoted by $f_{\tau \rightarrow t}$ and the backward optical flow by $f_{t \rightarrow \tau}$. Let the coordinates of a designated pixel be p . The new coordinates of the translated pixel in the subsequent frame, using optical flow, are computed as follows:

$$\hat{\mathbf{p}} = \mathbf{p} + f_{\tau \rightarrow t}(\mathbf{p}) \quad (8)$$

Next, we calculate the distance moved by this designated pixel in the horizontal and vertical directions according to the following equations:

$$\Delta = \|f_{\tau \rightarrow t}(\mathbf{p}) + f_{t \rightarrow \tau}(\hat{\mathbf{p}})\| \quad (9)$$

If the observed proportion of pixels η_{observed} exceeding the distance discrepancy is more than a predefined threshold η , we conclude that a major occlusion has occurred; otherwise, the occlusion is minor. The observed proportion of such pixels η_{observed} is calculated as:

$$\eta_{\text{observed}} = \frac{1}{H \times W} \sum_{i=1}^{H \times W} \mathbf{1}(\Delta_i > \epsilon) \quad (10)$$

where the denominator represents the total number of pixels in the image.

We disregard the gaze points for frames $\{I_\tau\}$ where there is major occlusion with respect to the image frame I_t and we represent this filtering using the function o_τ . If the occlusion is minor, the appropriate gaze points $\{g_\tau\}$ for all $\tau \in [t - \delta, t]$ are then translated into their new coordinates using the forward optical flow, and collected for the formation of heatmap H_t and subsequently.

As mentioned above, the idea is that if there is a major occlusion, the difference between the distance traversed by a pixel during forward optical flow, and the distance traversed by the translated pixel during backward optical flow will be significantly greater than in cases where the occlusion is minor. Optical flow was calculated using the implementation of the RAFT model developed by Teed and Deng (2020). The hyperparameter ϵ is the threshold distance, which was set to 20, whereas η is the threshold proportion of pixels that have exceeded the occlusion limit, set to 0.60. A brief overview of the process can also be seen in Algorithm 2.

Algorithm 2 Occlusion-Aware Gaze Point Aggregation

Input: Image sequence $\{I_\tau\}$ for $\tau \in [t - \delta, t]$,
 Gaze points $\{g_\tau\}$ at each timestamp τ ,
 Optical flow model (RAFT),
 Thresholds $\epsilon = 20$, $\eta = 0.60$

Output: Aggregated heatmap H_t

```

10 Initialize  $H_t$  as empty
11 for  $\tau = t - \delta$  to  $t$  do
12   Compute forward flow  $f_{\tau \rightarrow t}$  and backward flow  $f_{t \rightarrow \tau}$ 
13   for pixel  $\mathbf{p}$  in  $I_\tau$  do
14      $\hat{\mathbf{p}} = \mathbf{p} + f_{\tau \rightarrow t}(\mathbf{p})$ 
15      $\Delta = \|f_{\tau \rightarrow t}(\mathbf{p}) + f_{t \rightarrow \tau}(\hat{\mathbf{p}})\|$ 
16     Mark  $\mathbf{p}$  as occluded if  $\Delta > \epsilon$ 
17   Compute occlusion ratio:
18      $\eta_{\text{observed}} = \frac{1}{H \times W} \sum \mathbf{1}(\Delta > \epsilon)$ 
19   if  $\eta_{\text{observed}} > \eta$  then
20     Discard gaze point  $g_\tau$  (major occlusion)
21   else
22     Translate  $g_\tau$  to  $g_{\tau \rightarrow t}$  using  $f_{\tau \rightarrow t}$ 
23     Add Gaussian heatmap from  $g_{\tau \rightarrow t}$  to  $H_t$ 
24 Normalize  $H_t$ ;
25 return  $H_t$ 

```

6.4 Model Overview and Components

In our study, we initially employ the open-source version of the Flamingo model (Awadalla et al., 2023), based on the original Flamingo architecture proposed by Alayrac et al. (2022). Flamingo is a VLM designed to process interleaved image-text inputs, featuring a pre-trained vision encoder, a trainable Perceiver Resampler for constructing fixed-length visual tokens, and a cross-attention language decoder. To extend its capabilities for egocentric video understanding, we introduce a gaze-regularized attention mechanism that is placed before the Perceiver Resampler, enabling alignment between model attention and human gaze distributions.

Building on this design, we generalize our gaze-regularization framework to other transformer based VLMs, including LLaVA, InternVL, and LaViLa, by modifying their input pipelines as required to accept a sequence of image frames instead of a single frame. Our core idea remains consistent across

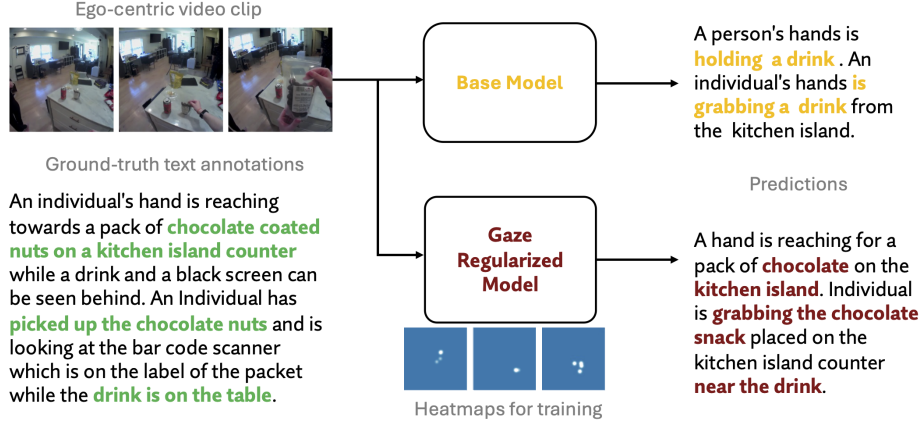


Figure 6: **Simple Illustration of Activity Understanding.** The model receives a sequence of egocentric frames as input and generates a fine-grained prediction of the current activity. While the base model misidentifies the object being picked up ('a drink'), the gaze-regularized model correctly predicts 'chocolate' by leveraging attention aligned with human gaze during training. Importantly, spatial gaze heatmaps are used only to guide attention during training and are not used at inference. Predicted annotations are shown on the right, with ground-truth descriptions below.

models: a gaze-based attention regularization mechanism is applied prior to fusing visual tokens with the language decoder, modulating attention to emphasize gaze-informed spatial regions.

It is important to note that while we adopt the architectures of models like LLaVA and InternVL, we do not use their original pretraining paradigms or loss functions. As such, the performance reported for these models should not be interpreted as a ranking of their full potential. Rather, our intention is to demonstrate the generalizability and impact of gaze regularization on different VLM architectures under a consistent training setup.

Inspired in part by the transformer-based egocentric gaze modeling approach of Lai et al. (2024), our method uses gaze heatmaps only during training to guide attention. During inference, the models rely solely on RGB image inputs, making them suitable for deployment scenarios where gaze data is unavailable. A simple illustration of this setup, along with example outputs for activity understanding, is shown in Figure 6.

To further ensure robustness, we incorporate an occlusion-check module based on optical flow consistency that filters unreliable gaze points during temporal aggregation. This improves the fidelity of gaze supervision and ensures that only visually relevant regions guide the model’s attention. The resulting framework enables gaze-guided fine-grained predictions and current activity understanding across multiple VLMs, while maintaining modularity and scalability.

6.5 Other Experiments

In this section, we present a series of ablation studies that explore the core design choices in our gaze-regularized framework. We begin by defining the evaluation metrics used to measure semantic alignment between predicted and ground-truth activity descriptions. Next, we conduct a sanity-check experiment to evaluate whether existing pretrained VLMs are capable of future activity prediction in egocentric settings.

We then analyze how the size and temporal density of gaze points affect model performance by varying size and number of gaze points used for heatmap aggregation. To disentangle performance gains due to architecture versus gaze supervision, we augment the base model with additional self-attention layers but without gaze regularization. We also test whether providing gaze coordinates in textual form (rather than as spatial heatmaps) yields comparable results.

Finally, we examine the effect of the observation horizon. Most of the ablations are conducted using the OpenFlamingo architecture as the base model, as it was the model initially used to get some


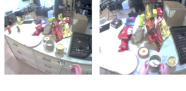
Future Frames	Ground Truth	Predictions(base)	Predictions (ours)
	The camera wearer is going to converse with the seated person in front with a deck of cards and a mobile phone screen visible in front of them. The individual is going to read the red book in front of her, possibly after being asked by the camera-wearer.	The individual has a TV remote with a game visible on the table and individual in front. The seated person is in the background in front of the card game with a book.	The camera wearer is going to talk to the seated person with the mobile phone visible. The seated person is going to read a red book which is near a deck.
	The camera wearer is going to open a plastic packet with a paper and a deck of cards inside and the person in a blue shirt will move in the living room. The individual is going to remove a deck of cards from a plastic packet.	The individual is opening a paper in the living room with motion in the background. The individual is removing a paper inside a plastic	The individual will open a deck of cards as blue shirt will move inside the living room. The camera wearer is going to remove cards from plastic.
	An individual's hand will reach towards a pack of chocolate coated nuts on a kitchen island counter while a drink and a black screen can be seen behind. An individual is going to pick up the chocolate nuts and will look at the bar code scanner which is on the label of the packet while the drink is on the table.	A person's hands will hold a drink. An individual's hand is going to grab a drink from the kitchen island	A hand will reach for a pack of chocolate on the kitchen island. Individual is going to grab the chocolate snack placed on the kitchen island counter near the drink.
	The camera wearer will pass cards to the person in the background with the headset. The card game will continue as the player places the cards on the table.	A person's hands will pass cards. The card came continues player will place on the table.	A camera wearer will pass cards to the person with the headset. The card came continues player will place card on the table.
	The individual's hand is going to reach for a brown container in front of the white cup and below some glasses in the bottom shelf of the cabinet. The individual is going to grab the brown container possibly full of sugar from the shelf.	The individual's hand will reach for container in the bottom shelf in the cabinet with glasses. The individual hand will grab the container of sugar.	The individual hand is reaching for a brown container below some glasses in the bottom shelf of the cabinet. The individual will grab the brown container of sugar from the shelf.
	The camera wearer will move towards the kitchen island with a spoon in hand. The individual is going to use the spoon to get some sugar from a container on the table full of snacks and a beverage at the end of the table.	The camera wearer will reach for container on the kitchen counter. The individual will grab some snacks on the table and beverage on the counter.	The individual is moving to the kitchen island with a spoon. The individual will grab the sugar from container using the spoon from the table full of snacks and a beverage on the counter.

Figure 7: **Extended Qualitative Results of Future Activity Prediction.** Examples comparing predictions from the base model and our gaze-regularized model against ground-truth annotations and actual future frames. The gaze-regularized model correctly predicts specific objects and actions (e.g., “picking up chocolate”) while the base model makes less accurate predictions (e.g., incorrectly predicting “a drink”). In addition, the gaze regularized model also gives finer-grained predictions by correctly predicting the color (e.g., “brown container”), objects surrounding the main object (e.g., “below some glasses”), which the base model fails to do in the predictions. By highlighting such properties, giving instructions to other humans or machines becomes easier, especially if there are multiple such objects involved. Key words in the predictions are highlighted to emphasize differences in prediction specificity and accuracy.

preliminary results and intuition about how to proceed further. Once the framework was validated, we generalized it to other architectures, as discussed in the main paper.

6.5.1 Evaluation Metrics

For evaluation, we propose using a semantic transformer (Reimers and Gurevych, 2019) to provide a quantifiable score that compares the generated output with the ground-truth action text, along with the ROUGE-L scores for some tables. This scoring system is designed to ensure that the model is not penalized for semantically equivalent phrasings. For example, ‘Football is being played by people’ and ‘People are playing football’ convey the same meaning and should yield similar scores. At the same time, the system should penalize outputs with nonsensical or incoherent word order. While some ablation tables in the appendix report both semantic and ROUGE-L scores for completeness, we omitted ROUGE-L from the main paper due to space constraints and because it exhibited highly correlated trends with the semantic scores. This allowed us to present a more concise evaluation while still conveying the key performance improvements

6.5.2 Evaluating Pretrained VLMs for Future Activity Prediction

Pretrained VLMs such as InternVL and OpenFlamingo have demonstrated impressive performance on a variety of tasks, including visual question answering, image captioning, and multi-modal reasoning. However, these models have not been explicitly designed or optimized for future activity prediction. To assess whether such models can be directly applied to short-horizon future prediction, we conduct a sanity-check experiment: we provide each VLM with a sequence of image frames and evaluate its ability to generate predictions about future actions.

Table 7: Effect of the size of the gaze point overlays on model accuracy

MODEL	SEMANTIC SCORE (\uparrow)
INTERNVL-2-1B	0.1572
INTERNVL-2-2B	0.1601
INTERNVL-2.5-1B	0.1596
OPENFLAMINGO-3B	0.1810
OPENFLAMINGO-4B	0.1878
OUR METHOD	0.7505

While this evaluation is not entirely fair, given differences in training objectives, data domains, and task formulation, it offers a useful diagnostic of how well existing models transfer to our setup. As shown in Table 7, current VLMs exhibit limited performance on our test set. This may stem from domain shift or the lack of temporal modeling in their pretraining. Nonetheless, the results highlight a promising opportunity for future work to extend VLM capabilities toward temporally grounded tasks like short-horizon future activity prediction.

6.5.3 Impact of Gaussian Overlay Size in Singular vs. Aggregated Gaze Models

In our exploration of gaze-regularized models, we investigated two primary strategies for representing gaze: (i) the *singular gaze model*, which uses a single gaze point per frame to form a spatial heatmap, and (ii) the *aggregated gaze model*, which combines gaze points from a temporal window with occlusion filtering.

To assess the role of spatial spread in these representations, we varied the standard deviation (σ) of the Gaussian kernel used to generate the gaze heatmap for both models. In the singular gaze model, increasing σ which expands the influence region around each gaze point, led to modest performance improvements. This suggests that slightly broadening the gaze-induced attention might help the model focus on semantically meaningful areas rather than overly localized pixels.

In contrast, increasing the Gaussian smoothing in the aggregated gaze model had minimal impact. As shown in Table 8, its performance remained nearly unchanged. This indicates that the strength of the aggregated model stems from its temporal diversity and occlusion-aware gaze selection, rather than from the spatial extent of individual gaze contributions.

These experiments were initially conducted using the OpenFlamingo architecture, which served as our primary reference during early ablation studies.

Overall, these findings highlight an important distinction: while spatial smoothing can benefit isolated gaze representations, temporally aggregated gaze signals already encode spatial redundancy via multiple temporally aligned points. Future work may investigate how the interplay between Gaussian spread (σ), image resolution, and aggregation window size affects performance.

Table 8: Effect of the size of the gaze point overlays on model accuracy

MODEL	σ	SCORE(\uparrow)	F-SCORE
SINGULAR	20	0.6211	0.4327
SINGULAR (LARGER OVERLAYS)	50	0.6512	0.4463
AGGREGATED	20	0.7505	0.5173
AGGREGATED (LARGER OVERLAYS)	50	0.7436	0.5034

6.5.4 Impact of Using More Gaze Points in Temporal Aggregation

To explore the impact of using a larger number of points (or a longer aggregation duration), we trained an aggregated gaze model with the openflamingo architecture utilizing 12 frames instead of 6 (and hence the aggregation time δ is 400 ms). The results of this experiment are provided in Table 9. This minimal improvement is likely due to the occlusion checks already filtering out less informative points, reducing the marginal utility of including more frames and limiting the number of usable

gaze points. In cases with minimal occlusion, the slight decrease in performance can be compared to that observed in an aggregated gaze model with larger overlays, where performance also decreased slightly. This result suggests that maybe utilizing an excessive number of gaze points could confuse the model. However, as this explanation is currently based on intuition rather than extensive analysis, this experiment was not included in the main paper.

Table 9: Comparison when more aggregated points are used for the gaze-regularized model

GAZE POINTS	SCORE(↑)	F-SCORE(↑)
6	0.7505	0.5173
12	0.7398	0.4971

6.5.5 Impact of Occlusion Filtering on Model Performance

In the aggregated gaze model, gaze points are collected within a specified time interval δ (200 ms). However, in dynamic environments, aggregating gaze points from the interval $[t - \delta, t]$ may introduce inaccuracies due to changes in the scene or camera movement. To mitigate this, we introduced an occlusion check to ensure that only relevant gaze points are aggregated. More details about the occlusion check method can be found in Sec. 6.3 in the Appendix. To evaluate the impact of this adjustment, we conducted experiments comparing models with and without the occlusion check in the future prediction task. As shown in Table 10, the openflamingo implementation of our framework incorporating the occlusion check slightly outperforms the one without it. The difference in the evaluation metrics can be attributed to the fact that only relevant and accurate gaze points are considered, which reduces noise and prevents the model from being confused by irrelevant data.

Table 10: Effect of using occlusion filtering during training

MODEL	SCORE	F-SCORE (↑)
GAZE-REG (W/O OCCL.)	0.7298	0.4800
GAZE-REG (W/ OCCL.)	0.7505	0.5173

6.5.6 Can Gaze Coordinates Alone Improve Model Performance?

In our studies, we converted gaze data from coordinate text form into heatmaps, which were then utilised by our gaze regularizer during training time. This transformation allows the regularizer to highlight important visual regions and aligns more closely with how humans perceive spatial information (Laeng et al., 2014). To conduct a sanity check and assess whether using gaze data in visual form is more suitable than using gaze in text form, we trained a gaze-regularized model that utilizes gaze coordinates as text input. Our results indicated that using gaze data in text form improved performance compared to the base model without gaze. However, it still fell short of the performance achieved by our model (as shown in Table 11). This highlights the importance of not just including gaze data, but representing it in a specific format such that it can be used aligned to modulate the spatial nature of model attention.

Table 11: Effect of using gaze information in text form and comparison with other models

MODEL	SCORE(↑)	F-SCORE (↑)
BASE	0.6525	0.4318
AGGREGATED GAZE(IN TEXT FORM)	0.7021	0.4630
AGGREGATED GAZE	0.7505	0.5405

6.5.7 Addition of Self Attention Blocks to the Base Model

To ensure a fair comparison, we evaluate whether the performance improvements in our gaze-regularized model stem solely from its architectural enhancements, specifically, the extra attention

layers that operate on a global query derived from the full image sequence. To test this, we augment the base OpenFlamingo model by adding two self-attention layers and providing it with the same global query, but without applying any gaze-based regularization. This allows us to isolate the effect of architectural changes from the influence of gaze supervision. As shown in Table 12, the modified base model does show improved performance compared to the original baseline. However, it still falls short of the performance achieved by the gaze-regularized models. This suggests that while incorporating global queries and self-attention blocks enhances the model’s capacity to aggregate contextual information, it is the gaze-guided regularization that ultimately drives stronger alignment with human attention and increases model performance.

Table 12: Comparison of base model with attention block against gaze regularized models

MODEL	SCORE(↑)	F-SCORE(↑)
BASE	0.6525	0.4318
BASE (W SELF-ATTENTION)	0.6701	0.4393
GAZE-REGULARIZED	0.7505	0.5173

6.5.8 Training Base VLM with Gaze-Embedded Images

As a sanity check, we evaluate what happens when baseline models are trained on gaze-embedded RGB images, where gaze heatmaps are directly superimposed onto the original RGB frames. This setup allows us to assess whether performance improvements could arise simply from exposing the model to gaze-like visual cues, without any explicit use of gaze during training or inference, as is the case in our gaze-regularized framework, which does not take gaze as input. All models were trained under the same conditions, using identical data splits. As shown in Table 13, using gaze-embedded inputs yields a modest performance gain over standard RGB inputs. However, this improvement remains well below the performance achieved by our gaze-regularized models. These results suggest that the benefits of our approach stem from the explicit use of gaze as a training-time regularizer, rather than from simply incorporating gaze like patterns in the input images.

Table 13: Comparison for egocentric event prediction when gaze-overlaid images are provided to base VLM.

MODEL	SCORE(↑)	F-SCORE
BASE W RGB IMAGE	0.6525	0.4318
BASE W GAZE-EMBEDDED IMAGE	0.6873	0.4435
GAZE REGULARIZED	0.7505	0.5173

6.5.9 Effect of Temporal Gaze Aggregation on Performance

As described in the main paper, we construct gaze heatmaps by first transforming individual gaze points into spatial maps and then aggregating them over a short temporal window. This aggregation is accompanied by occlusion filtering to ensure that only visually valid gaze points contribute to the final heatmap used for attention regularization.

To understand the specific contribution of temporal aggregation, we compare performance against a baseline that uses only a single gaze point at each timestep—referred to as the *singular gaze model*. In this case, the gaze heatmap H_t at time t is defined as:

$$H_t = m_t = \pi(G_\sigma * \mathbf{1}(g_t)), \quad (11)$$

is an indicator function with value 1 at position g_t and 0 elsewhere, G_σ is a Gaussian kernel with standard deviation σ , $*$ denotes convolution, and π represents a normalization so the heatmap sums to 1.

Our experiments show that aggregating gaze points over a short interval (e.g., 200 milliseconds) consistently improves performance over the singular gaze variant. This improvement arises because temporal aggregation captures the stable and temporal structure of visual attention across



Figure 8: **Illustration of singular and aggregated gaze heatmaps.** For the *singular* gaze heatmap, only the gaze point associated with the final heatmap is utilized. On the other hand, for the *aggregated* gaze heatmap, gaze points over an interval $\delta = 200ms$ are collected to form the aggregated heatmap.

frames, whereas single gaze points can be often noisy, particularly if sampled during saccades or fleeting fixations. In addition, over-regularization of attention to a small space (pertaining to a singular overlaid point) might not always be beneficial, as shown by the performance degradation as compared to the baseline model where no gaze regularization is used.

By integrating gaze information from multiple frames (and by including an occlusion filtering mechanism), the model gains a richer and more reliable supervisory signal that better aligns its attention with human intent. This temporal consistency enables more accurate understanding of ongoing actions and better prediction of upcoming behavior. A simplified illustration comparing singular and temporally aggregated gaze heatmaps is shown in Figure 8.

Table 14: Semantic performance on future prediction and activity understanding tasks comparing singular and aggregated gaze models. Gains indicate the absolute improvement from using aggregated gaze.

MODEL	FUTURE PREDICTION			ACTIVITY UNDERSTANDING		
	SINGLE	AGGREGATED	GAIN	SINGLE	AGGREGATED	GAIN
OPENFLAMINGO	0.6512	0.7505	+9.9%	0.6913	0.7848	+9.4%
MODIFIED OPENFLAMINGO	0.6024	0.7200	+11.8%	0.6631	0.7300	+6.7%
LAVILA NARRATOR	0.6037	0.6924	+8.9%	0.6364	0.7268	+9.0%
INTERNVL	0.6223	0.7318	+10.9%	0.6813	0.7404	+5.9%
OPENLLAVA	0.5814	0.6771	+9.6%	0.6500	0.7027	+5.3%

6.5.10 Change in Observation Horizon for Future Prediction

In the main paper, we reported results for future prediction tasks using an anticipation horizon of $\tau_a = 2$ seconds and an observation horizon of $\tau_o = 5$ seconds. To further analyze the model’s behavior, we conducted an additional experiment by reducing the observation horizon to $\tau_o = 3$ seconds i.e., using a shorter input sequence to predict the same future window. Interestingly, this reduced observation window led to slightly improved performance for some models, while maintaining consistent overall gains when gaze regularization is used as opposed to a baseline model, as shown in Table 15.

This suggests that even with limited visual context, our gaze-regularized models can effectively anticipate future actions. Interestingly, while humans leverage visual working memory for short-term planning, this memory may not be effectively simulated over longer temporal horizons in our current framework. As a result, tracking the visual cues may become harder as the anticipation window grows, leading to reduced predictive accuracy. While this remains a speculative explanation, it motivates further investigation into integrating memory inspired mechanisms into our gaze-inspired framework. Accordingly, we present these exploratory results in the appendix rather than in the main paper as the explanation is intuitive and not experimentally investigated.

Table 15: Semantic performance on future prediction task with reduced observation horizon (3 seconds). The aggregated gaze model continues to outperform the base model across all architectures.

MODEL	SINGLE	AGGREGATED	GAIN
OPENFLAMINGO	0.6616	0.7565	+9.5%
MODIFIED OPENFLAMINGO	0.6200	0.7281	+10.8%
LAVILA NARRATOR	0.6144	0.6966	+8.2%
INTERNL	0.6301	0.7279	+9.8%
OPENLLAVA	0.5851	0.6731	+8.8%

6.5.11 Activity Understanding for Static Image

We examine whether a static scene, independent of sequential context, is sufficient for the gaze-regularized models to understand an ongoing activity, noting that VLMs are often used in static-scene setting. While a single image lacks temporal context, it contains visual cues and object affordances that hint at probable actions. Unlike future prediction, which benefits from evolving gaze patterns, static scene understanding relies mainly on visible objects. If gaze is beneficial, it should highlight key regions that aid activity understanding. To test this, we compare the base and gaze-regularized models in generating fine-grained descriptions from single images, isolating the impact of gaze without temporal cues. Results from Table 16 show that the gaze-regularized model achieves a modest 1.3 % improvement over the base model, indicating that gaze has moderate impact on scene understanding in single-image settings. However, its full potential is realized in tasks where evolving gaze patterns provide richer contextual cues. Hence for the activity understanding tasks in the main paper, we provide a long enough observation horizon $\tau_o = 3s$ such that the activity can be understood, as well as to utilize the evolving temporal gaze patterns.

Table 16: Comparison between base model and gaze-regularized models (with different regularization coefficients) for activity understanding tasks.

MODEL	SCORE	F-SCORE
BASE	0.6897	0.4432
GAZE-REGULARIZED	0.7014	0.4611

6.5.12 Effect of Using Sequence-Level vs. Frame-Level Queries for Attention Computation

Table 17: Impact of using sequence-level queries on semantic prediction for activity understanding. Using global queries from the entire input sequence improves performance on both in-distribution and out-of-distribution data.

MODEL VARIANT	SCORE (TEST)	SCORE (OOD)
GAZE-REGULARIZED (W/O GLOBAL QUERY)	0.7284	0.6902
GAZE-REGULARIZED (WITH GLOBAL QUERY)	0.7505	0.7305

In our main framework, we compute a global query using the entire input image sequence, rather than generating separate queries for each individual frame. This choice supports better temporal generalization. If frame-level queries are used, the model processes each image independently, without awareness of the broader activity context. As a result, the attention mechanism may become overly dependent on visual patterns specific to individual frames.

This becomes particularly problematic in tasks like future prediction, where the same frame could occur in different activity sequences. Without access to temporal context, a frame-level query cannot distinguish between these scenarios, increasing the risk that the model memorizes attention patterns instead of also relying learning task-relevant cues. In contrast, a sequence-level (global) query captures the dynamics of the entire clip, encouraging the model to focus on features that reflect

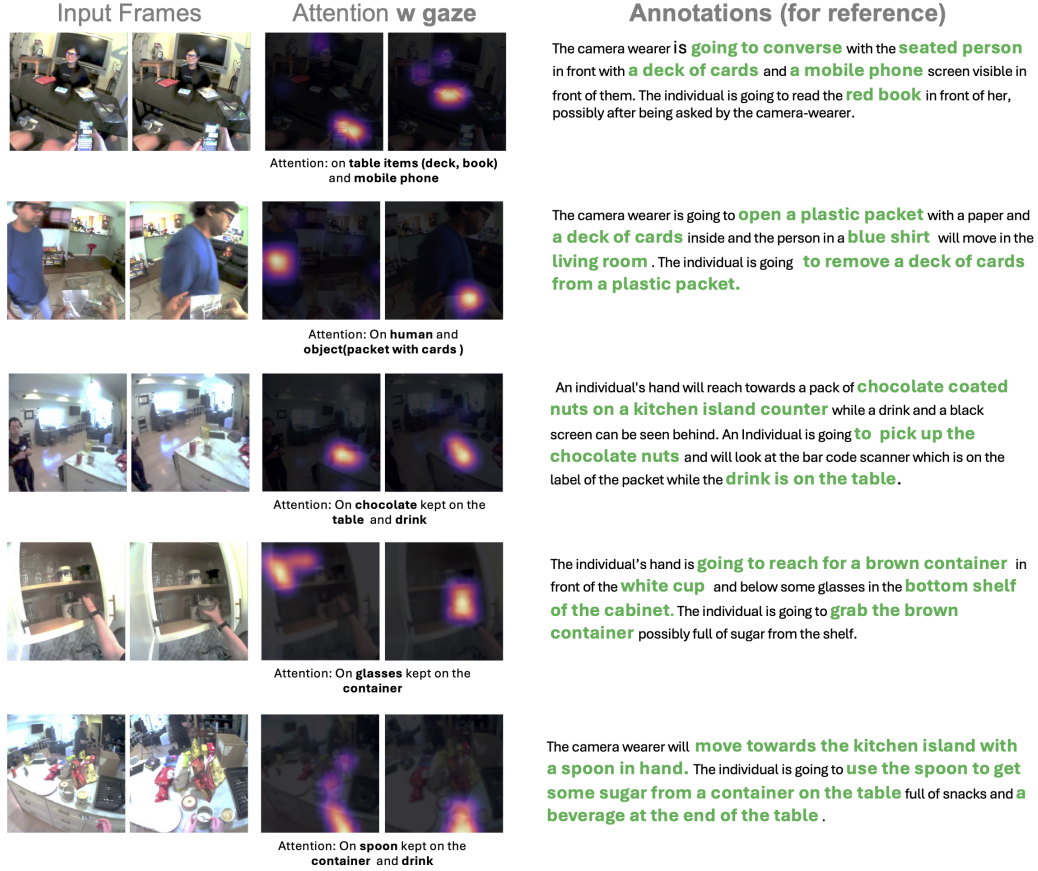


Figure 9: **Extended Comparison of Attention Maps.** This figure compares attention distributions from the base model (without gaze regularization) and our gaze-regularized model. Our method produces attention maps that are more semantically aligned with human gaze, leading to better focus on task-relevant regions. For instance, in the second-to-last row, the model attends to both the glasses and the container—objects emphasized in the ground truth annotation. In the final row, attention is correctly directed toward the hand and the container, improving the model’s ability to interpret and predict the ongoing activity.

the underlying activity rather than frame-specific objects. From Table 17, the results support our intuition: incorporating global queries not only improves semantic scores but also results in a smaller performance drop on out-of-distribution data compared to using frame-wise queries.

6.6 Discussion and Limitations

In the following section, we briefly discuss some limitations of our current framework and outline promising directions for future work.

While our gaze-regularized framework offers strong improvements in egocentric understanding tasks, certain constraints remain. To maintain computational efficiency, we downsample videos to 1 frame per second. Although this preserves high-level temporal patterns, it may limit sensitivity to rapid or transient actions. Nonetheless, we find that the current performance and predictions capture egocentric behavior understanding in majority of the scenarios.

A common challenge in egocentric vision is the presence of motion blur or poor visual quality. In sequences where most frames are heavily blurred, even gaze-regularized models struggle to generate accurate predictions, as both visual and gaze cues become less informative. Figure 10 illustrates such a case. Even though we tried to identify most of such ground truth annotations and mitigate them, some cases still persist and this is one of the limitations.

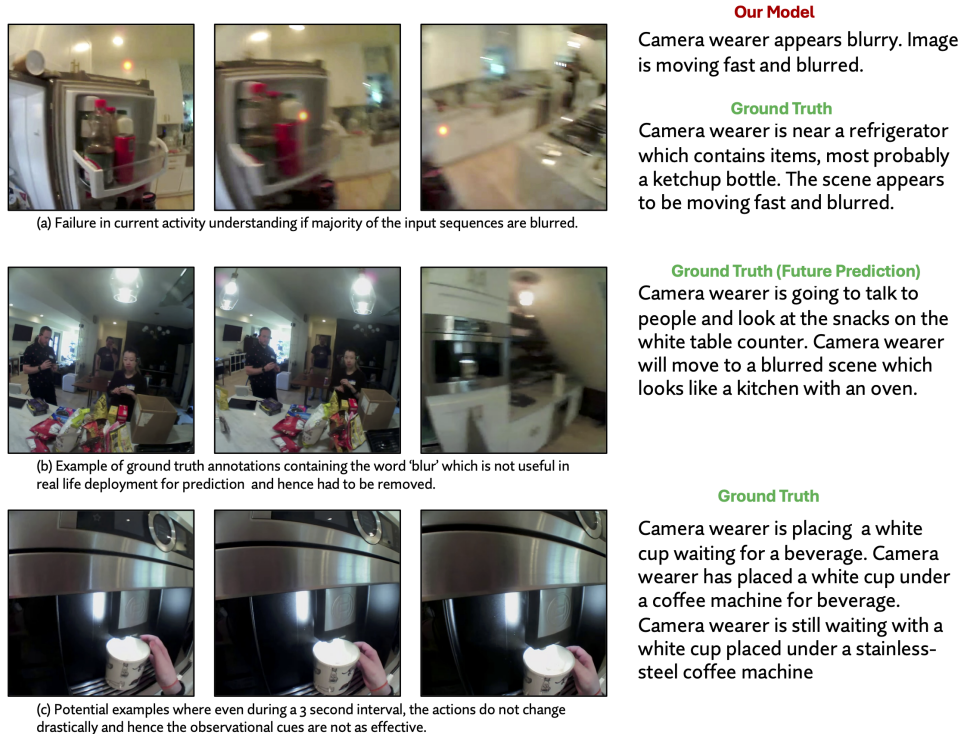


Figure 10: Example failure case and qualitative observations. *Top:* When the majority of frames in the input sequence are significantly blurred, even gaze-regularized models struggle to produce accurate predictions. Although the ground-truth annotations preserve contextual intent, the lack of clear visual input limits the model’s ability to reason about ongoing actions. *Middle:* Examples of image-text pairs that were excluded during dataset construction because they contained words like “blur” or “unclear,” which offer little utility for activity understanding or future prediction. We aimed to minimize the inclusion of such samples. *Bottom:* A case where the actions remain largely consistent across the sequence, making the activity easier to infer regardless of the modeling approach.

866 While our occlusion-aware gaze filtering improves robustness, the reliability of gaze supervision still
 867 depends on accurate calibration. Misalignment in eye-tracking data can lead to suboptimal heatmaps,
 868 which in turn may misguide the attention regularizer. For training the models from scratch using
 869 custom data, it is necessary to ensure that the gaze calibration used for collecting training data is in
 870 tune with the camera wearer.

871 Our findings offer encouragement and also point towards the need for deeper exploration of how gaze
 872 aligns with visual representations in VLMs. For example, recent observations in Bolya et al. (2025)
 873 suggest that the final-layer features of vision encoders may not always be ideal for all downstream
 874 tasks. This raises an interesting parallel: just as feature selection matters in model design, it is worth
 875 asking whether certain stages of gaze processing e.g., early fixations vs. late context scanning, align
 876 better with specific types of reasoning, such as classification, visual search, or question answering.

877 This leads to a promising direction for interdisciplinary collaboration. Psychological research has
 878 long shown that gaze behavior is task-dependent: humans scan scenes differently when searching
 879 for an object versus answering a question. Extending this insight to VLMs could help define how
 880 attention should behave in task-specific settings. By incorporating gaze collection protocols tailored
 881 to distinct tasks, we may be able to design models that not only align better with human attention but
 882 also adapt their internal focus in a task-aware manner.

883 In summary, our framework provides a strong foundation for integrating gaze as a training signal
 884 in egocentric VLMs. Future work can build on this by exploring task-specific attention modeling,
 885 leveraging insights from both vision science and cognitive psychology, and investigating how different
 886 forms of gaze data interact with visual representation choices across diverse downstream tasks

6.7 Training and Evaluation details

Both the base model and the gaze-regularized model were trained using two NVIDIA A800 40GB GPU cards. For initial experiments, we used the OpenFlamingo architecture to develop and evaluate our approach. The base OpenFlamingo model required approximately 36–38 hours to train, while the corresponding gaze-regularized version took around 50 hours. This additional time is due to the added regularization computation and the insertion of gaze-aligned attention blocks.

Training was conducted with a batch size of 32 and a learning rate of 7×10^{-5} over 10 epochs. The vision encoders were kept frozen and pre-trained. To accelerate training, we employed Fully Sharded Data Parallel (FSDP), which efficiently distributes model parameters and gradients across GPUs, improving memory usage and speed. Data loading was managed using the WebDataset loader, with datasets converted to .tar format for compatibility with both WebDataset and FSDP.

After validating our method with OpenFlamingo, we extended the same training pipeline to other architectures such as InternVL, LaViLa, and OpenLLaVA. In each case, the integration of our gaze-regularized component followed the same principle: it was inserted immediately after the visual encoder and before the language decoder, allowing for modular modulation of attention without disrupting the rest of the model architecture.

To give a comparison in terms of model complexity, the base model for openflamingo has approximately 944 million parameters. The gaze-regularized model with 2 additional attention blocks, which is the best performing model, has approximately 966 million parameters.

For evaluation, models were assessed using the semantic transformer (SBERT) developed by Reimers and Gurevych (2019), along with ROUGE-L scores. Regarding the runtime, we would like to clarify that the evaluation was conducted using RGB images for the base model and the gaze-regularized model. On average, the base model took approximately 1.7-1.8 seconds to process a sequence while the latter took 2.2 - 2.3 seconds . However, this runtime does not include the time taken for image loading or pre-processing overhead. The testing was also conducted on a NVIDIA A800 40GB GPU card.