Fast Localization and Slow Classification of Mouse Behavior Prediction using Sparse Video with Trajectory Estimation

Audrey Douglas
aadougl@umich.edu
University of Michigan
USA

ABSTRACT

Manual labeling of animal behavior, such as in mice studies, is time-consuming, inconsistent, and limits large-scale analysis. Machine learning-enabled labeling could solve this bottleneck in animal behavior studies. Although recent advances in video understanding with transformer-based vision-language models (VLMs) enable richer temporal embeddings, they remain computationally infeasible for continuous video and lose information over long videos, with the problem beginning in videos as short as a minute. To address this, we propose a two-stage temporal action localization pipeline for mouse behavior recognition. In the first stage, features, either from pose estimation or frame embeddings, are used to identify candidate action segments. In the second stage, these segments are processed with the Qwen2.5-VL video VLM to obtain action segment embeddings, which are then classified using either a neural network (for actions with many labels) or clustering (for actions with few labels). We evaluate our method with pose embedding features on both a supervised and few shot learning mouse behavior dataset. By combining low-cost action localization with high-accuracy VLM-based classification, our approach aims to reduce computational overhead while maintaining performance, offering a scalable method for predicting mouse behavior in long or continuous video recordings.

KEYWORDS

Video Understanding, Animal Behavior, Pose Estimation, Temporal Action Localization

1 INTRODUCTION

Traditionally, studies involving mice behavior include manual action labeling. This is time intensive, limits data collection, and involves large variation among different annotators [14]. These limitations extend to human action recognition as well [5]. This has caused an influx of attempts to automatically label data, which often rely on pose estimation, reducing complex visual data to just a few numbers, or per frame video analysis, which misses out on complex actions involving many frames [14] [15]. To go beyond these methods, would involve embedding multiple video frames at once, which provide a more complex action representation. Until recently, processing many video frames at once in an effective temporal manner was impossible [1]. With recent advances in transformer-based video understanding we can now embed many seconds of video at once [1]. However, this process is still too computationally expensive to run on an entire video, and current methods of compacting video embeddings leave out information [1]. Given that mice are not always performing relevant actions, we should focus only on particularly salient parts of the video. To deal with this, our method proposes using temporal action localization techniques, along with state of the art video video language models (VLMs). Specifically, we will do this in two parts. First, we use pose estimation features to identify key video frames where an action is likely to be performed, and second, we feed chunks of video around these frames to a VLM to get embeddings, which are sent to a vanilla neural network to classify actions. Compared to methods that process and reduce the whole video, this remains low in computation because many irrelevant frames will not be embedded while utilizing state of the art techniques. The key here is using a two-stage anchor mechanism where we train the first stage to identify mice actions, but heavily rely on an off-the-shelf VLM trained for multiple purposes in the second stage, which should allow for more apt one shot learning. The ability to use a low cost action localization technique with a high cost action identification technique can allow the method to be feasible over long and even continuous video instances. As mentioned, action recognition is highly relevant in many areas, and high accuracy at low cost is desirable behavior.

2 PROBLEM DEFINITION

Video understanding and temporal action recognition are improving rapidly, with their improvements usually moving in tandem. We aim to see how a two stage pipeline, with a low-cost complete analysis of the video, which can point to areas that require high computation to properly identify actions. Harnessing recent video VLMs for action recognition, which incorporate frames temporally, we aim to see if we can improve animal behavior prediction, even if this is very different from what off-the-self models are trained on. Within this work, we look at identifying behavior in multi-agent mouse interactions.

Using extracted features from videos of mice, along with video clips, we aim to predict what actions the mice are taking. From end to end, our aim is to use video frames and pose estimation to predict whether mice are performing attack, investigation, mount, or other behavior at every frame of the video. The dataset we use [14] has video, pose estimation, and per frame video labels. To tackle this problem, we will be using a temporal action localization pipeline in two stages. The first stage aims to go from extracted video features (pose estimation) to predict whether or not an action is occurring (0 is yes, 1 is no). This gives us a per frame prediction of if an action is occurring. Once actions have been localized, video clips predicted to have actions (the localized areas) are fed into a pretrained video VLM to get embeddings. These will be inputted into a vanilla NN or clustered to identify actions.

All tasks are supervised. The action localization is performed with labels and the second phase dealing with video embeddings will involve a supervised vanilla NN and KNN clustering. Our second task involves few-shot learning.

3 RELATED WORK

3.1 Animal Behavior

Observing and studying human behavior is immensely important for various fields. This need well-known within the computer vision community, with a large focus in recent years on action recognition. Improving performance on humans does not always correlate to animals though, which present their own unique sets of behaviors [14]. Identifying these behaviors can be useful for tasks that aid humans. For instance, it would be useful to automatically track the behavior of mice in drug studies.

Born out of this desire to automate mouse behavior labeling, the Caltech Mouse Social Interactions dataset captures multi-agent animal behavior. It provides three types of data, well-labeled frame-by-frame annotation, inherently inconsistent annotations done by many annotators, and sparsely labeled unique actions for few-shot tasks [14]. This presents a unique opportunity to utilize pose estimation for action recognition in mice because multi-agent behavior inherently involves more movement [14]. In addition, researchers are interested specifically in multi-agent behavior [14].

This body of work has grown beyond mice to other animals, in the MABe22 dataset [15]. Here, they expand the data to include metadata like experimental conditions and time of day, which are relevant factors to predict, while being low cost labels to generate [15]. For the sake of this study, we focus on mice and labeled behavior data due to video VLMs being trained in scene representation and description rather than nonvisual differences. There are also many other animal behavior datasets within the community, with others involving OpenBehavior, a mouse dataset with many action across many labs [6], 3D-ZeF, a zebra fish tracking dataset [10], and Fly v Fly, which tracks fruit fly social interaction [4].

Our work does not aim to add to datasets or find new behavior, but interleaves supervised temporal action localization with few shot action recognition to focus in and learn a domain, in this case mice, while allowing for generalization, in this case actions.

3.2 Action Recognition

The specific task of action recognition has been one of the first tackled by the video understanding community due to the short time frames some actions take place over [20]. As the community has progressed, actions over longer time frames and in more dynamic environments have been studied.

3.2.1 Video Understanding. One of the most popular video understanding datasets is Ego4d, a first-person action-based dataset [5]. Ranging from clips as short as 1 minute to as long as an hour, Ego4d encompasses a wide range of actions. Actions range from cooking to playing sports, so the complexity of tasks varies greatly.

The task of fully comprehending these videos is far from complete though. Recent advances in computing power and transformer sequence length input have given video VLMs the ability to process longer chunks of video without aggregation. With the advent of deep learning, video understanding took off with CNN architectures [20]. Most recently, VLMs have been at the forefront of video understanding. The VLM originated with CLIP, a model that encodes text and language separately, but learns to represent text captions near their image counterparts [12]. This allows text and image representations to be interleaved in these models. A video is simply a series of images stacked together, much like how text is a series of words stacked together. As such, transformer advances in language follow in vision, particularly in token length and history consolidation.

Regardless, these techniques are still often not enough to process the video data for the action labelling task. For instance, with animal behavior data you may have a near infinite stream of data available. With video VLMs alone, such a task would be infeasible due to computational scaling.

There are many approaches to dealing with longer videos, which can be divided into resolution and architecture changes. The resolution changes focus on memory consolidation, such as summarizing past embedding tokens [2], reducing resolution with pooling [8], and masked attention [13]. These models become very efficient, but often lack an ability to derive nuance in videos. There are also a number of methods focused on architecture changes, such as sparse attention mechanisms [19] and logarithmic attention mechanisms [3]. These have been less adopted than their competitors and have not yet bled into the language vision model space, focusing solely on language.

This work breaks up videos into 5 second segments which are then processable with modern techniques. In many ways these can still be classified as longer videos.

Many state of the art models are already implementing these techniques, but we will focus on the open source Qwen2.5-VL model that utilizes dynamic resolution and sliding windows [1].

3.2.2 Temporal Action Localization. To process long videos without a computationally intensive model to learn all of the feature representations, we use temporal action localization. This task has been around as long as action recognition tasks, where a model aims to find when an action occurs in a video, and then classify it [16]. These methods can be understood as two paradigms: anchor and classification methods. Anchor mechanisms rely on the existence of temporal anchors, which are the start and end of when an action takes place. These can be chosen though methods that simultaneously choose likely action boundaries and classify the action (one-stage) [9], those that first find likely action boundaries then use a different model to classify (two-stage) [17] and those that choose their own action boundaries that are not predefined (anchor-free) [7]. In this study we will be focusing on a two-stage anchor mechanism, since this will allow us to use video VLMs for classification. Two-stage approaches allow module flexibility, but are more sensitive to network settings due to multiple pieces interacting [16].

In contrast to anchor methods which make predictions of if an action is happening between anchors, classification methods directly compute action instances as their output. These methods usually involve a frame by frame classification [11], where thresholds are grouped, or a model's input being the proposed time sequences [18].

Our work will utilize a two-stage classification method. This will allow us to harness classical action localization techniques while

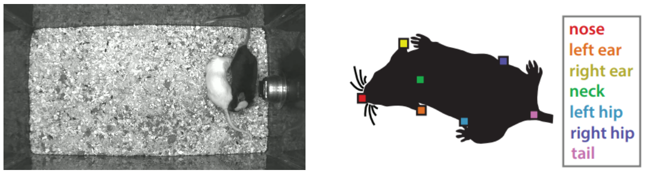

Figure 1: Left: Here is an example video frame from the dataset. One mouse is white and the other is black, which helps differentiation, and the video is shot directly overhead. Right: These are the key points extracted for each mouse by the creators of the dataset. [14]

bringing in new transformer models for classification, which as a whole pipeline, are just recently being explored in this domain.

4 METHOD

Within this work we use a two-stage anchor mechanism for temporal action localization of multiagent mouse behavior.

4.1 Dataset

We use the Caltech Mouse Social Interaction Dataset [14]. It includes long form video, pose estimation, and manually annotated action labels. Within this work we will be using Task 1 and Task 3 from this dataset, which will be further referred to as the supervised and few-shot task. In the supervised task, Task 1, a single human annotator labeled every frame of 70 videos with one of 4 actions: close investigation, attack, mount, and other. In the few-shot task, Task 3, 7 more actions are labeled in approximately 3 videos each. This is still a supervised learning task, but there is not enough data to finetune a model on the data, so a few-shot approach is taken in stage two. Videos are shot at 60 frames per second. All videos in the dataset have machine extracted pose estimation features which are specified in Figure **??**. These are represented as 28 dimensional vectors (7 keypoints, 2 mice, 2 dimensions). Per frame action labels are all created by a single annotator, so this study will mimic their annotation style. For training the first part of our network, labels will be reduced to 0, action, and 1, no action, where the class 1 is entirely frames labeled as other.

4.2 Temporal Action Localization

We are using a two stage anchor mechanism. This involves a first pass which predicts when actions occur. If an action is predicted/sustained, this is selected as an action proposal. The next pass involves refining anchor selection and categorizing the action proposed within the time frame.

4.2.1 Stage One. For our temporal proposal subnet we will use a modified version of the baseline proposed in [14]. We will use a convolutional neural network architecture over the temporal direction, with intermediate channel sizes $[128, 64, 32]$ and filter size 5.

Since whole video analysis is not possible and transformers are very costly, we train on extracted features. In particular we will use the 28 dimensional mouse pose vectors from the dataset [14]. Our model inputs 100 frames of pose estimation (50 before and 50 after, skipping every other frame) and outputs a binary classification. This stage can be seen in Figure 2.

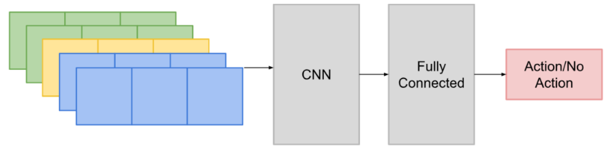

Figure 2: Stage one of the pipeline takes in many frames of mice poses and runs them through a CNN and fully connected layer to predict action versus no action.

Our model is trained on the supervised task, reduced to binary classification, which allows for generalizing to actions occurring versus not. This will be able to be used in the few-shot learning task.

After having a per frame prediction of if an action is occurring, our action start proposals will be taken as times when the prediction flips from no action, 1, to an action 0, where an action persists for at least $\frac{1}{3}$ of the next 300 frames. The action proposal will end 300 frames later, when the video ends, or when the next action proposal occurs. This allows for dynamic action proposal starts, allowing us to capture all actions.

4.2.2 Stage Two. From stage one, we now have temporal segments to analyze for actions. We will use Qwen2.5-VL, an open source video vision language model [1]. Qwen2.5-VL accepts video and processes it temporally, utilizing dynamic resolution and sliding window methods to process large inputs, allowing for richer embeddings of videos [1]. Each video segment will be run through the model with the prompt "what are the two mice doing in this video?". This should prime the model to create an embedding relevant to what we are looking for. Rather than using the text output, we will use the final hidden vector state for further analysis.

Two methods of classification are used in this study. For the supervised case, we use a vanilla neural network with these embeddings to predict classification. A no behavior class is included as well. This version of the architecture can be seen in Figure 3 To identify actions for the few shot task, where we only have a few example videos, videos are processed through the pipeline, and a k-nearest neighbors algorithm is used on the embeddings used to select the nearest action. In this case, the original 3 actions are still included in the network's training, since this is what our temporal segmentation algorithm will be finetuned with.

5 EXPERIMENTS

5.1 Dimensions

The Experiments are broken down between the supervised and few-shot learning task.

5.1.1 Task. The pipeline is run on both the supervised and few-shot learning mice behavior data. As mentioned earlier, the supervised learning task has 4 behavior labels across 70 videos [14]. The few-shot learning task looks at 7 more behaviors, but is much more limited in annotations, and involves some rare actions [14]. Stage one of the anchor mechanism will always be trained on the

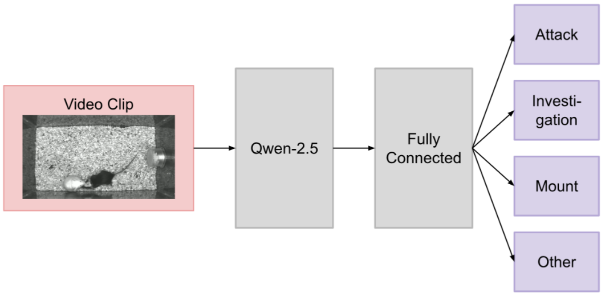

Figure 3: Stage two of the pipeline passes 5 second video clips into Qwen2.5-VL, and runs the outputted embeddings through a fully connected layer to predict actions.

supervised task data, which is only trying to find where actions exist. For the broader pipeline, everything will remain unchanged except for the final classification method. For the supervised task, we use a vanilla NN and for the few-shot task, we use clustering.

5.2 Success and Requirements

5.2.1 Metrics. Given that our labels align with that of a standard classification task, to align with the creators of the dataset, we have opted for our evaluation metrics to be class-average F1 and Mean Average Precision score [14]. A successful study will involve an improvement in either of these scores for our algorithm.

5.2.2 Data. The dataset being used is the Caltech Mouse Social Interactions Dataset [14]. This contains mouse videos, in MP4 format of varying sizes (<1000 MB), and pose data, which contains per frame 28-dimensional vectors (7 key points x 2 mice x 2 dimensions). For all labeled data, behaviors are labeled per frame.

6 RESULTS

6.1 Action Localization

To quickly understand what sufficient performance on our data means, we can view how frequently actions are taken in Table 1. Here we can see that 'other' (which we will consider as no action) is the most common, but not by an overwhelming amount. This is a good baseline for how a naive model can do. That is to say, our model ought to achieve at least 0.63 accuracy.

Attack	Investigation	Mount	Other
0.03	0.29	0.05	0.63

Table 1: Distribution of mouse action for the supervised task

We can view our action localization technique as the upper bound on how good our model pipeline can perform. Since we will only send the sections of videos that we think have actions to stage two, our best possible accuracy will be that of stage one. In this instance we only have 1 class, so we will use accuracy, precision, and recall as our metrics. We are able to achieve very high testing accuracy of 0.93, along with 0.88 precision and 0.96 recall. These are all sufficiently high that the identification of temporally relevant segments should not pose an issue for solving this problem.

From this stage, 824 and 431 clips were selected out of the training and testing data respectively. For reference in the testing data there are 873 possible 5 second non overlapping clips.

6.2 Embedding and Classification

The action localization clips were then ran through Qwen2.5-VL to obtain 1,536 dimensional embeddings. Along with that came language output which was not used to train the model. Regardless, there are notable language outputs (Table 2), which will be used to infer what the model focuses on in the discussion.

Nondescript	'In the video, the two mice are seen running around a small cage.'
Detailed	'In the video, the two mice are seen moving around a small cage or enclosure. One mouse is black and the other is white. The black mouse is moving around the cage more actively than the white mouse. The white mouse is also moving around the cage, but it appears to be more stationary than the black mouse. The two mice seem to be exploring their environment and may be searching for food or other objects.'
No Response	"I'm sorry, but I cannot provide an answer to your question as there is no video or image available for me to analyze. Please provide me with a video or image so that I can assist you better."

Table 2: Categories of Qwen2.5-VL model language output

After the embeddings for all of the temporal clips were created, we then classified each clip to have an action. Using our classifier, the vanilla NN, for the label on each clip, our model achieved a test accuracy of 0.47, F1 score of 0.33, recall score of 0.37, and precision score of 0.34.

6.3 End-to-End

Now that we have action classification predictions out of stage two along with temporal action predictions from stage one, we can merge our results together to get final predictions for all frames of the video, as seen in Table 3). This incurs both error picked up from stage one and two of training. We also have per action predictions in Figure 4. Here performance was very high for the 'other' action and decently high for the 'investigation' action, but performed poorly in both the 'mount' and 'attack' behavior, likely due to data imbalance.

Model	F1 Score	MAP
Baseline	0.79	0.85
Pose + VLM	0.52	0.52

Table 3: Comparison of our model to the baseline for end-to-end action prediction

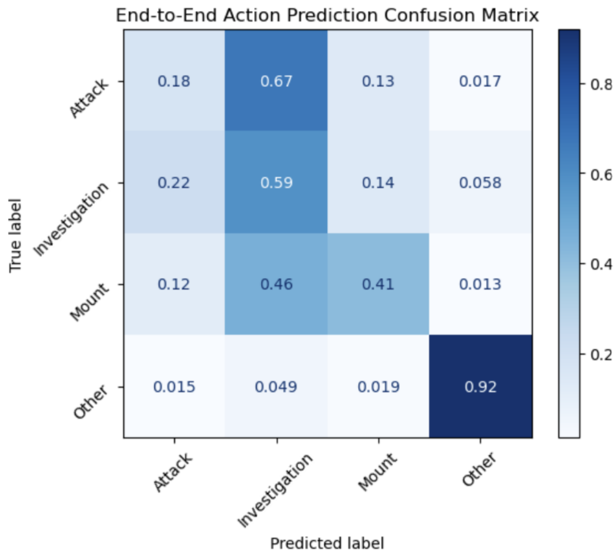

Figure 4: Confusion matrix showing model performance across actions. There is a tendency to predict investigation and very high accuracy for properly identifying the action 'other'.

6.4 Few-Shot Learning

For the few-shot learning classification task, the model performed very poorly. In clustering with all of the samples from the supervised task, overall, the few-shot behaviors had precision of 0.01 and recall of 0.25. To further see this drop in performance, the action 'sniff-face' was only identified in 1 out of 27 clips. This was considerably worse than the supervised task, and is probably a result of unbalanced classes in our data. Compared to the 824 embeddings for the first 3 actions, these 7 actions had 231 embeddings. These poor predictions were also likely to occur due to the curse of dimensionality, with our embeddings having dimension 1536.

7 DISCUSSION

7.0.1 Action localization outperforms action prediction. As seen from our results, in action localization it is easy to achieve high performance. In this case, we already knew decent results were achievable with this architecture when predicting actions, so it was likely that action localization would achieve better results. Here our MAP score as 0.88 which is higher, though not by much, than that achieved by the baseline action classification model that had 0.85. Our accuracy was very good though at 0.93. This means that most of our downstream results were not negatively affected by this part of the model. Given that 0.63 of the results were also not one of the actions to be labeled, it was also very important for this part of the network to perform well. In many ways, this portion of the network accounts for a bulk of the full model's positive results.

7.0.2 Fine grained processing is important for action labeling. One of the major pitfalls of our model was that by trying to reduce the computation on the VLM, ie. not running it for every frame of each video, we were forced to chunk pieces of the video together. This intuitively seems correct since if you are performing an action in one frame you are likely also performing it in the next, but given the discontinuity of or models action prediction, it could predict an action sequence such as [1,0,1,0,1,1,0,1], it was hard to find a proper threshold for what counted as a continuous action. To deal with this, longer chunks of video, up to 5 seconds, were put into the VLM. The window context of our stage one network is also around 5 seconds, but that process is trained per frame, where as our VLM doesn't have enough examples to properly distinguish actions from the beginning or end of a clip.

Further, this chunking left messy projections at the end of stage two from video clip predictions to frame predictions. If, for instance, a clip from t_1 was action 1 and the next clip t_2 was action 2, all values recognized to have an action between t_1 and t_2 were set to action 1. This could have been as short as 2 frames or as long as the rest of the video. This inherently doesn't allow for seamless changing of actions, and can lead to the outlier action prediction, ex. a single 0 in a field of 1s, to get classified as an unrelated action. A possible fix for this problem would be to also have set action localization points to train the later network on, ie set video clips. This would leave room for what was previously considered an action to now be classified as other.

7.0.3 VLMs are too language forward. What was originally considered to be the backbone of this works contribution fell short on the fact that VLMs are very language bound. VLMs are trained on video captions, which are distinctly different from labeled mouse behavior. For captioning videos, it is often more important to know what is in the scene rather than the details of what happens in it. Furthermore, captions are not centered around the fact that we could use them to distinguish pictures, they are focused on aiding our knowledge of what is in a picture.

We used video embeddings, so to some extent, the language output is not exactly what we were looking at, but given that the language output is a direct filter down of the video embeddings, it should be a good proxy for what is occurring in the video. In our results we see 3 types of responses. The nondescript response, where the model points out 2 mice and then stops, the detailed response, where movement is mentioned in some form, and no response, where the model seems to think there is no video.

Of the first 2 forms of output, nondescript is clearly less helpful, but this early ending could be a result of model temperature causing early stopping. I would also say that while the detailed description includes more information, it in no way aids in the ability to determine what action is taking place.

The no response option boils down to a few different problems with the data. I think the first comes from overly short videos. Sometimes the actions were found to be near the end of the video, and when this happened we got very short video clips, which led to strange output. Further Qwen2.5-VL is a censored model, as almost all LLMs are. I believe that when it considered something to be mounting behavior this is what was outputted. In that case though, having an embedding that could identify this would still be useful.

7.0.4 Few-shot learning finds unbalance in embedding space. Qwen2.5-VL is created for language output and not embeddings. This means that doing any type of clustering or machine learning on top of the model is not intended to perform well because the embedding space is not being optimized through training. There are some models of Qwen2.5-VL and other LLMs which are made for embeddings, but these lack the video capabilities we wanted. Given that we were working with the embeddings, I think this added some problems to the end of our model pipeline.

We can see this through our few shot learning example. The new action, sniff-face was embedded only 31 times across 3 videos, and in a space of 824 other embeddings, it was almost never nearest to its own action. This likely implies that embedding space is either very scattered or very close. Given our language outputs, the latter seems likely. Interestingly, the nearest neighbor to the sniff-face action was never another sniff-face action.

8 CONCLUSION

What was expected to have state of the art results actually ended up showing many of the limitations of VLMs in action localization. They themselves are not great temporally, and if feeding in a multi-action video, will not properly reconstruct action sequences. As such, video clips need to be fed in, but this leads to the problem of where they start where they end, and what part of the clip is actually being identified to have an action. To alleviate some of this we added an action localization network, which performed very well, but even that was not enough to properly deal with the chunking problem. This problem was likely due to the fact that many VLMs, including Qwen2.5-VL, aren't designed for spatial tasks, they are designed for language. Their captioning ability is accurate but no precise and their generalizabiliy leads to the need for censoring data, which in scenarios where these things are necessary can lead to messy embeddings, which can't be properly used to distinguish video features. I do think general purpose models will eventually make their way into the non human action localization and recognition space, but I think more work is needed on their video context, sequence reconstruction, and spatial understanding.

9 APPENDIX

9.1 Additional Resources

More information and resources on this project can be found at https://audreyadouglas.github.io/mouse_action_rec/. The code can be found at https://github.com/audreyadouglas/mouse_action_rec/tree/main.

9.2 ChatGPT

Language models were purely used to aid in the construction of the abstract. The output was not taken at face value.
(1) Model I Used: ChatGPT.
(2) Provide an abstract for "insert the rest of the proposal here".
(3) I find language models to be very good at summarization and given that I have been in the depths of this abstract and reading related papers a lot is going on in my head. I think getting a new perspective on a summary is good and helps keep it high level.
(4) It included a few irrelevant sentences for an abstract and over-mentioned the dataset used but was otherwise fine.

REFERENCES

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025).

[2] Dibyadip Chatterjee, Edoardo Remelli, Yale Song, Bugra Tekin, Abhay Mittal, Bharat Bhatnagar, Necati Cihan CamgÃ¶z, Shreyas Hampali, Eric Sauser, Shugao Ma, et al. 2025. Memory-efficient Streaming VideoLLMs for Real-time Procedural Video Understanding. arXiv preprint arXiv:2504.13915 (2025).

[3] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens. arXiv preprint arXiv:2307.02486 (2023).

[4] Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. 2014. Detecting social actions of fruit flies. In European Conference on Computer Vision. Springer, 772–787.

[5] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 18995–19012.

[6] Mark Laubach. 2021. Collaborative Research: CIBR: The OpenBehavior Project. NSF Award Number 1948181. Directorate for Biological Sciences 19, 1948181 (2021), 48181.

[7] Luxuan Li, Tao Kong, Fuchun Sun, and Huaping Liu. 2019. Deep point-wise prediction for action temporal proposal. In International conference on neural information processing. Springer, 475–487.

[8] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. Mvitv2: Improved multiscale vision transformers for classification and detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 4804–4814.

[9] Tianwei Lin, Xu Zhao, and Zheng Shou. 2017. Single shot temporal action detection. In Proceedings of the 25th ACM international conference on Multimedia. 988–996.

[10] Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B Moeslund. 2020. 3d-zef: A 3d zebrafish tracking benchmark dataset. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2426–2436.

[11] AJ Piergiovanni and Michael Ryoo. 2019. Temporal gaussian mixture layer for videos. In International Conference on Machine learning. PMLR, 5152–5161.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In International conference on machine learning. PmLR, 8748–8763.

[13] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. 2023. Hiera: A hierarchical vision transformer without the bells-and-whistles. In International conference on machine learning. PMLR, 29441–29454.

[14] Jennifer J Sun, Tomomi Karigo, Dipam Chakraborty, Sharada P Mohanty, Benjamin Wild, Quan Sun, Chen Chen, David J Anderson, Pietro Perona, Yisong Yue, et al. 2021. The multi-agent behavior dataset: Mouse dyadic social interactions. Advances in neural information processing systems 2021, DB1 (2021), 1.

[15] Jennifer J Sun, Markus Marks, Andrew Wesley Ulmer, Dipam Chakraborty, Brian Geuther, Edward Hayes, Heng Jia, Vivek Kumar, Sebastian Oleszko, Zachary Partridge, et al. 2023. Mabe22: A multi-species multi-task benchmark for learned representations of behavior. In International Conference on Machine Learning. PMLR, 32936–32990.

[16] Binglu Wang, Yongqiang Zhao, Le Yang, Teng Long, and Xuelong Li. 2023. Temporal action localization in the deep learning era: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 46, 4 (2023), 2171–2190.

[17] Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-c3d: Region convolutional 3d network for temporal activity detection. In Proceedings of the IEEE international conference on computer vision. 5783–5792.

[18] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. 2020. G-tad: Sub-graph localization for temporal action detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 10156–10165.

[19] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. Advances in neural information processing systems 33 (2020), 17283–17297.

[20] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. 2020. A comprehensive study of deep video action recognition. arXiv preprint arXiv:2012.06567 (2020).