Exploring the Impact of Model Parameters and Components on Video Saliency Prediction with Foundation Models

Morteza Moradi[1], Mohammad Moradi[1], Francesco Rundo[2], Concetto Spampinato[1], Ali Borji[3*], and Simone Palazzo[1*]

[1] University of Catania, Catania, Italy
{morteza.moradi, mohammad.moradi}@phd.unict.it,
{concetto.spampinato, simone.palazzo}@unict.it
[2] STMicrolectronics, ADG Central R & D, Catania, Italy
francesco.rundo@st.com
[3] Quintic AI, San Francisco, CA, USA
ali.borji@quintic.ai

Abstract. As a companion to the ICPR 2024 accepted paper "SalFoM: Dynamic Saliency Prediction with Video Foundation Models", this work investigates how various model parameters and components impact its performance. Since SalFoM represents the first effort of its kind in this field, the additional experiments presented here are designed to provide insights into the application of video foundation models for dynamic saliency prediction. This is achieved by exploring different aspects of the model's architecture and the use of large video models. Additionally, this work analyzes the impact of various strategies for defining training objectives on the model's learning capabilities and overall performance. The code is available at https://github.com/mr17m/SalFoM—Video-Saliency-Prediction.

Keywords: Video Saliency Prediction · Video Foundation Model · Reproducibility

1 Introduction

The advent of video foundation models (VFMs) has opened up numerous opportunities for various video understanding tasks. However, as with any new paradigm, it takes time for the community to become familiar with how these models work, how they can be adopted, and how to best leverage their capabilities for specific tasks that were traditionally performed using other approaches.

In particular, when it comes to utilizing VFMs for video saliency prediction (VSP), there has been a lack of published research—aside from our model, SalFoM [7]—leaving the community with limited knowledge on how such large models can be employed. The promising performance of SalFoM has set a strong

[*] Equal supervision

precedent for the design of VFM-based VSP models. With the goal of providing the research community with the key insights needed to use large video models for VSP, this companion paper explores the influence of various parameters and components on model performance. By modifying the original architecture of SalFoM and exploring different components—specifically, variations of the VFM-based feature encoder, namely UnMasked Teacher (UMT) [5], and the 3D shifted window-based Transformer modules [6] used in the feature decoding stage—we aim to demonstrate how the vast amount of features extracted from a video foundation model can be leveraged to enhance video saliency prediction performance. Additionally, we will examine how different variations of the 3D shifted window-based Transformer modules in the feature decoding stage affect the model's overall performance.

Moreover, since SalFoM leverages spatio-temporal transformers in the feature decoding stage, selecting the optimal number of transformer layers during this phase is crucial. The goal is to strike a balance between maximizing performance and minimizing computational overhead. Achieving this balance could pave the way for more efficient designs of feature decoders that harness the power of spatio-temporal transformers. By carefully adjusting the number of layers, researchers can unlock new insights into how spatio-temporal architectures can be fine-tuned for improved accuracy and resource efficiency in a variety of applications.

In a different aspect, the objectives upon which a VFM-based dynamic saliency prediction model is trained play a crucial role in its effectiveness. Therefore, this study examines the performance of the SalFoM model trained with various loss functions, utilizing the most commonly used evaluation metrics. By exploring these different training approaches, we aim to gain insights into how the choice of loss function impacts the model's predictive accuracy and overall performance in dynamic saliency prediction tasks. These insights not only clarify our design choices in the implementation of SalFoM but also enhance reproducibility and serve as a best practice guide for developing future models in the field.

We conducted all of our experiments on the DHF1K [9] dataset. We used 10% of the training set as the validation set and utilized the publicly available validation set as the test set for evaluating model performance, as the ground truth for the official test set is not publicly available. We adopted the same experimental setup as SalFoM [7], utilizing the Adam optimizer [4] for gradient descent, with an initial learning rate of 10^{-5}.

The remainder of the paper is organized as follows: In Section 2 and its subsections, we investigate the impact of modifying the configurations of the decoder (including the transformer and CNN modules in the feature decoding stages) and UMT as the feature encoder of the model. In Section 3, we analyze the effect of using different training objectives on the model's performance.

2 Model Components

2.1 3D shifted window-based transformer modules

One of the key components in the feature decoding stage of SalFoM is the use of 3D shifted window-based Transformer modules in the first intermediate feature decoding branch. These modules are designed to capture spatio-temporal features, which are crucial for effective visual information processing. While spatio-temporal Transformers excel at capturing complex visual data, their performance tends to scale with the number of attention layers incorporated. However, increasing the number of layers also introduces additional computational overhead and can result in overfitting due to the large number of parameters involved.

To better understand the trade-offs associated with the number of attention layers in the 3D shifted window-based Transformer modules within SalFoM, we conducted a series of experiments. These experiments varied the depth of the attention layers to evaluate their impact on model performance, computational efficiency, and the risk of overfitting.

In SalFoM's current decoding stage, each Transformer module consists of 6 layers. To investigate how varying the number of layers affects model performance, we conducted a series of experiments by adjusting the number of layers to 2, 4, 8, and 10.

The performance results for each model variant, corresponding to these different layer configurations, are presented in Table 1. These findings provide valuable insights into the balance between increasing model depth for enhanced feature extraction and managing the associated computational cost and overfitting, helping to inform the optimal architectural design of SalFoM.

Table 1. Evaluation of the impact of varying layer counts in SalFoM's Transformer-based feature decoder on the DHF1K validation set. The highest score for each metric is indicated in bold.

Transformer Modules Depth	CC	NSS	SIM	AUC-J	Size (MB)	♯Params(M)
Depth 2	**0.565**	3.174	0.417	0.923	1560	399.5
Depth 4	0.561	3.168	0.435	0.924	1567	401.2
Depth 6	**0.565**	**3.299**	0.436	**0.928**	1574	402.9
Depth 8	0.560	3.197	**0.441**	0.924	1580	404.6
Depth 10	0.562	3.194	0.436	0.924	1587	406.3

As shown in the table, when comparing different transformer module depths to the main SalFoM model (which uses a depth of 6), a few patterns become evident. The model with a depth of 2 achieves CC scores comparable to the main SalFoM model, but falls short across all other evaluation metrics. Similarly, the model with a depth of 4 also underperforms in every metric compared to the reference model with a depth of 6. In the case of a model with a depth of 8, although the SIM metric surpasses that of the reference model, the other metrics are lower. This improvement in SIM comes at the cost of increased model size and a higher parameter count. Finally, for a model with a depth of 10, only the SIM score matches that of the main SalFoM model, while all other metrics show a decline, coupled with a further increase in model size and parameters.

It has been observed that incorporating a larger number of layers, i.e. more than 6 layers, in the transformer modules during the decoding stage of a dynamic saliency prediction model does not enhance overall performance and incurs significant computational costs and may lead to diminishing performance gains, especially if the model is already sufficiently deep for the task. On the other hand, using a lower number of layers (less than 6 layers) in those modules results in a reduced parameter count and smaller model size, it negatively impacts the model's overall performance and can result in underfitting. This emphasizes the critical importance of finding the optimal configuration to ensure the model delivers its best performance.

As the model's complexity grows, the time and resources needed for both training and inference also increase. Therefore, it is important to strike a balance between model size and computational efficiency to achieve optimal performance without overloading computational resources.

2.2 CNN-based decoding branches

In the feature decoding stage of SalFoM, the second intermediate feature decoding branch—Dynamic Feature Decoding (DFD)—plays a crucial role by preventing abrupt down sampling of temporal information, thereby maintaining temporally-rich details and capturing intricate local features. As both the number of consecutive decoding layers and the approach to reducing temporal dimensions significantly influence the final saliency map reconstruction [8], this section examines the effects of incorporating an alternative type of 3D convolutional layer—specifically, the depth-wise separable 3D convolutional layer [10]—due to its reduced number of parameters and lower computational cost. Additionally, it explores the impact of varying the number of 3D convolutional layers within the second intermediate decoding branch on the overall network performance. This is explored by either adding extra layers or reducing the number of consecutive layers within this stage.

In one experiment, an additional 3D convolutional layer was added after each existing layer to maintain both the feature channels and temporal dimensions without any reduction. This setup aims to assess the effects of gradual temporal dimension reduction and the incorporation of additional layers in the feature decoding stage.

In a contrasting experiment, the impact of abrupt reductions in both temporal and channel dimensions was explored by halving the number of consecutive layers. In this configuration, the first layer reduces the channels from 1024 to 256 and halves the temporal dimension. The second layer further reduces the channels from 256 to 64, again halving the temporal dimension. Finally, the third layer decreases the channels from 64 to 16 and reduces the temporal dimension by a factor of four. This experiment not only investigates the consequences of sudden temporal dimension reduction but also explores the effects of sharp decreases in channel dimensions within the feature decoding stage.

In the feature decoding stage of SalFoM, the third intermediate feature decoding branch (Static Feature Decoding, SFD) is designed to abstract temporal effects and focus on spatial information. This section explores the impact of using a different type of 2D convolutional layers, as well as varying the number of layers within this intermediate branch. In this regard, in an experiment we employed 2D depth-wise separable convolutional layers [2] for each layer in this branch.

On the other hand, in an experiment we aimed to explore the effect of using additional consecutive 2D convolutional layers in this branch. So, we added additional 2D convolutional layers after the third, sixth, and ninth layers of the original model, ensuring that the added layers did not change the number of feature channels. In another experiment, we investigated the impact of decoding with fewer consecutive layers in this branch by eliminating the third, sixth, and ninth layers from the original network.

Table 2. Evaluation of the effects of different layer types and counts in SalFoM's DFD and SFD intermediate feature decoder branches. The highest score for each metric is highlighted in bold. The final row of the table shows the performance of the original version of SalFoM.

Model	CC	NSS	SIM	AUC-J	Size (MB)	♯Params(M)
DFD branch:						
3D depth-wise Conv Layers	0.463	2.463	0.329	0.901	1547	396.1
Extra Layers	0.559	3.188	0.433	0.923	1586	406.0
Fewer Layers	0.564	3.178	0.432	0.924	1564	400.4
SFD branch:						
2D depth-wise Conv Layers	0.463	2.484	0.341	0.897	1550	396.8
Extra Layers	**0.568**	3.192	**0.441**	0.924	1576	403.5
Fewer Layers	0.560	3.185	0.428	0.924	1571	402.3
SalFoM	0.565	**3.299**	0.436	**0.928**	1574	402.9

As shown in Table 2, while employing depth-wise convolutional layers in both the DFD and SFD branches reduces the number of model parameters, it not only fails to improve performance but also leads to significant performance degradation. Furthermore, increasing or decreasing the number of 3D convolutional layers in the DFD branch did not enhance the overall performance of the model. On the other hand in the case of SFD, using more 2D convolutional layers does outperform CC and SIM; however, the improvement is modest and comes with an increased number of parameters. Additionally, reducing the number of layers in this branch leads to a decline in overall performance.

2.3 Encoder layers

Given that video foundation models are typically large and contain a significant number of parameters, investigating the effects of varying layer counts in these models for feature encoding in dynamic saliency prediction tasks can provide insights for developing lighter models suitable for resource-intensive applications.

To investigate this, we conducted extensive experiments using different variants of the video foundational model, UMT, as the feature encoder for our dynamic saliency prediction task. Specifically, we used the large version of Unmasked Teacher (with 24 encoding layers) that processes 16 frames (UMT-L-16) in SalFoM. By adjusting the depth of the transformer layers in UMT-L-16, we aimed to identify the optimal configuration that balances performance and computational efficiency. Additionally, we performed similar experiments with the large version processing 8 frames (UMT-L-8) and the base version processing 8 frames (UMT-B-8) to explore the impact of input video clip length and transformer layer depth on the performance of video foundational models, as well as their impact on dynamic saliency prediction.

By altering the number of encoding layers in these models, we anticipate that a video foundational model-based feature encoder will show weak performance when the layer count is small. This is because a shallow encoder struggles to capture abstract patterns and high-level features, limiting its capacity to learn hierarchical representations. On the other hand, increasing the depth of the feature encoder is expected to improve its generalization capability and enhance its ability to model intricate details and long-range dependencies more effectively.

Table 3. The effect of utilizing video foundation models with varying numbers of layers as the feature encoder in the SalFoM dynamic saliency prediction model. The highest score for each metric is shown in bold.

Model's Encoder	**CC**	**NSS**	**SIM**	**AUC-J**	**Size (MB)**	**♯Params(M)**
UMT-B-8:						
Depth : 6	0.497	2.764	0.384	0.910	360.1	92.1
Depth : 12	0.527	2.935	0.400	0.915	526.2	135.0
UMT-L-8						
Depth : 6	0.475	2.604	0.352	0.903	596.4	152.7
Depth : 12	0.533	2.969	0.397	0.916	891.8	228.2
Depth : 18	0.547	3.100	0.426	0.920	1187.1	303.8
Depth : 24	0.552	3.169	0.418	0.924	1482.3	379.6
UMT-L-16						
Depth : 6	0.502	2.802	0.379	0.911	688	176.2
Depth : 12	0.542	3.062	0.412	0.920	983	251.8
Depth : 18	0.562	3.185	0.430	0.923	1280	327.3
Depth : 24	**0.565**	**3.299**	**0.436**	**0.928**	1574	402.9

In our experiments, we employed all the encoders pretrained on the Kinetics 400 dataset [3] as the feature encoder of our video saliency prediction model, i.e., SalFoM. As illustrated in Table 3, we found that the UMT-B-8 encoder, which operates with a depth of 6, outperforms the UMT-L-8 encoder, which has the same depth and frame count and also it even outperforms SIM when compared to the UMT-L-16 model that processes 16 frames. While larger video models like UMT-L generally demonstrate superior performance on action recognition tasks, e.g., Kinetics 400 dataset, this advantage is primarily due to their higher number of parameters and increased depth. However, when these models are required to function with fewer layers, their ability to utilize the extensive knowledge acquired during pretraining diminishes.

In contrast, the performance decline seen in smaller video foundation models— such as UMT-B-8, which are pretrained with fewer parameters on large datasets— when a number of its layers are removed for specific downstream tasks, is less significant. These models preserve a considerable amount of the knowledge gained during pretraining, enabling them to maintain relatively strong performance even with reduced complexity. This indicates that while larger models may perform exceptionally well in certain scenarios, their performance benefits can be diminished when some layers are removed. In contrast, smaller models demonstrate greater adaptability to variations in architecture. The same observation is evident at a depth of 12, where the performance of the UMT-B-8 is comparable to that of the UMT-L-8 and, notably, it even achieves a higher score in the SIM evaluation.

As the number of layers in the UMT-L-8 increases, the VSP performance improves correspondingly. When we examine the UMT-L-16 model, we find that its performance surpasses that of the UMT-L-8. This indicates that processing a greater number of frames enhances the model's ability to capture long-range dependencies among video frames, effectively compensating for the impact of having a lower number of layers. This suggests a positive correlation between model depth and performance, highlighting the importance of both the number of frames processed and the architectural depth in enhancing the model's ability to understand complex temporal relationships in video data. Overall, these findings underscore the importance of optimizing both frame count and layer depth to achieve superior performance in video saliency tasks.

3 Model's Training

To evaluate the effectiveness of optimization procedures for a video foundation model-based dynamic saliency prediction, this section will explore how various evaluation metrics influence the training process and the model's performance. We will analyze their impact on the overall efficacy of the dynamic saliency prediction model to identify the most effective strategies for enhancing accuracy and reliability in predicting dynamic saliency in videos.

While the SalFoM network is initially trained using the Correlation Coefficient (CC) and Kullback-Leibler divergence (KL) metrics, this study expands the evaluation by incorporating additional widely used metrics, including Normalized Scanpath Saliency (NSS) and Similarity (SIM), into the loss function. These metrics were chosen for their established effectiveness in assessing saliency map performance.

In a series of experiments, we trained the SalFoM network using each evaluation metric—CC, KL, SIM, and NSS—individually. This approach enables us to investigate the specific impact of each objective on the performance of dynamic saliency prediction. This comprehensive examination will deepen our understanding of the strengths and limitations of each metric, ultimately informing the optimization of the SalFoM network for dynamic saliency tasks. The definitions of each evaluation metric can be found in [1].

Table 4. Impact of different training objectives on the performance of a video foundation model-based dynamic saliency prediction. The highest score for each metric is shown in bold.

Loss Function	**CC**	**NSS**	**SIM**	**AUC-J**
CC	0.547	3.121	0.362	0.916
KL	0.564	3.192	**0.442**	0.925
SIM	0.310	1.948	0.234	0.806
NSS	0.556	3.139	0.375	0.918
CC+KL	**0.565**	**3.299**	0.436	**0.928**
CC+KL+SIM	0.557	3.141	0.437	0.921
CC+KL+SIM+NSS	0.558	3.160	0.431	0.921

It has been observed in Table 4 that using any single evaluation metric as the loss function fails to provide a sufficient objective for training the model. This approach negatively impacts learning, tuning, and ultimately deteriorates the model's overal performance. However, when both CC and KL are combined in the loss function, the model can be optimized more effectively, leading to improved performance. Conversely, when the number of metrics used in the loss function increases to three or four, the training process struggles to optimize the network across all metrics. Since each metric evaluates the model from a different perspective, the optimization process becomes unable to meet the expectations of all metrics simultaneously, leading to degraded model performance.

4 Conclusion

Reconducting extensive experiments to design VSP models based on video foundation models, as introduced in [7], requires substantial computational resources and is time-consuming. In this work, we aim to explore various aspects of implementation of SalFoM to facilitate further research in the field. Additionally, we seek to provide other researchers with the opportunity to uncover the underlying structure of the model's different components, particularly the leveraged video foundation model, without the need to run additional experiments not addressed in the original paper. This approach paves the way for both reproducibility of results and the development of new VSP models.

References

1. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? IEEE transactions on pattern analysis and machine intelligence **41**(3), 740–757 (2018)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
3. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR **abs/1705.06950** (2017)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: Towards training-efficient video foundation models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19948–19960 (October 2023)
6. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3202–3211 (2022)
7. Moradi, M., Moradi, M., Rundo, F., Spampinato, C., Borji, A., Palazzo, S.: Salfom: Dynamic saliency prediction with video foundation models. In: 27TH International Conference on Pattern Recognition (2024)
8. Moradi., M., Palazzo., S., Spampinato., C.: Transformer-based video saliency prediction with high temporal dimension decoding. In: VISAPP 2024. SCITEPRESS (2024). `https://doi.org/10.5220/0012422800003660`
9. Wang, W., Shen, J., Xie, J., Cheng, M.M., Ling, H., Borji, A.: Revisiting video saliency prediction in the deep learning era. IEEE transactions on pattern analysis and machine intelligence **43**(1), 220–237 (2019)
10. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018)