Siamese Masked Autoencoder: A Reproduction Under Resource Constraints

Alexander Weers **Edward Leander** **Ferran Sanchez Llado**

Abstract

This paper reproduces Siamese Masked Autoencoders (SiamMAE), a novel approach to object recognition and scene understanding in machine learning systems. The SiamMAE model, which combines the principles of Masked Autoencoder (MAE) and Siamese neural networks, uses a Vision Transformer (ViT) to use temporal information from videos to enhance object learning and other vision tasks. Our study, however, introduces a constraint of limited resources, providing insight into the challenges and solutions associated with such a limitation. We delve into two unique implementations: a self-implemented variant where everything is created from scratch and a from-MAE variant which starts from the public implementation of MAE, that is similar to how it is described in the paper. We compare and contrast the two approaches and note that the use of a pretrained MAE reduced the requirement for a large amount of data and computation. In the end, both implementations have successfully replicated the SiamMAE and we expanded on the paper by exploring multiple masking ratios for the siamese encoder and cross-self decoder variant. In conclusion, we were able to deepen our understanding of SiamMAE, MAE, VIT and Transformers, through our journey to replicate this paper.

1 Introduction

In the rapidly evolving field of computer vision, addressing the challenge of establishing correspondence between images or scenes, particularly in the presence of occlusions, viewpoint changes, and diverse object appearances, remains a focal point of research. Gupta *et al.* paper "Siamese Masked Autoencoders"[5] introduces an innovative extension of Masked Autoencoders (MAE)[6], designed to tackle the complex task of learning visual correspondence from videos. SiamMAE operates on pairs of randomly selected video frames, employing an asymmetric masking approach. These frames are independently processed by an encoder network, and a decoder, which comprises of a sequence of cross-attention layers. That is tasked with predicting missing patches in the future frame. The trained encoder from the SiamMAE outperforms "state-of-the-art self-supervised methods on video object segmentation, pose keypoint propagation, and semantic part propagation tasks"[5].
Our project consists of reproducing the SiamMAE architecture under heavy resource constraints, focusing on a qualitative analysis of the second frame being reconstructed and a quantitative analysis using video object segmentation via video label propagation [8, 9].

2 Related Work

The Siamese Masked AutoEncoder (SiamMAE)[5] is derived from the foundational structure of the Masked AutoEncoder (MAE)[6]. The MAE, in turn, is based on the architecture of the Vision Transformer (ViT)[3]. The foundation of Vision Transformers trace back to the broader class of general Transformers[11].
In SiamMAE, Gupta *et al.* randomly samples a "pair of video frames and randomly mask a huge

HT23 Deep Learning Advanced, DD2412, KTH.

fraction (95%) of patches of the future frame while leaving the past frame unchanged"[5]. Gupta *et al.* processes the two frames "independently by a siamese encoder parameterized by a ViT"[5] and the "decoder consists of a sequence of cross-attention layers and predicts missing patches in the future frame" [5]. The authors of SiamMAE compare their approach to many works, including VideoMAE [10] and DINO [2].
The original SiamMAE tested various masking ratios for the joint encoder and joint decoder setup. Therefore, our work expands on it by experimenting with the siamese encoder and cross-self decoder setup with a masking ratios of 50%, 75% and 95%. We focus on the task of video object segmentation for our quantitative results.

3 Methods

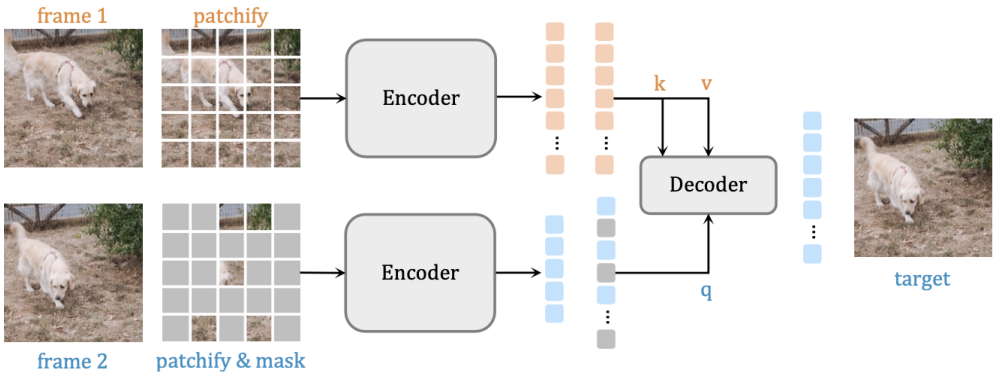


Figure 1: Schematic representation of the SiamMAE architecture [5]. Two frames, f_1 and a significantly masked variant of f_2 with a random frame gap, are extracted from a video and independently processed through the same encoder ϕ_{enc}. Cross-attention layers in the decoder ϕ_{dec} then try to reconstruct the unmasked version of f_2.

To create a training context in which the model is required to learn the concept of 'objects', Gupta *et al.* [5] implement a Masked Autoencoder [6] within a Siamese setting [1].
As part of the data loading process, two frames, f_1 and f_2, must be sampled from each video, with the condition that f_1 precedes f_2. Furthermore, the gap between these frames is randomly selected within a range of 4 to 48.
Since the underlying architecture is a Vision Transformer [3] it is necessary to first *patchify* the input frames f_1 and f_2, i.e. reshape them in a way to create $n_{\text{patch}} = \frac{w}{p} \cdot \frac{h}{p}$ patches, where p is the patch size.
The patches of f_1 are then passed through the encoder part to get features $g_1 = \phi_{\text{enc}}(f_1)$. The patches of f_2 are first masked randomly $f_2' = M(f_2; r_{\text{mask}})$ with a mask ration r_{mask}, in a way that only $1 - r_{\text{mask}}$ patches are kept. Those masked patches are then passed through the same encoder to get features $g_2' = \phi_{\text{enc}}(f_2')$. After that the previously masked patches are inserted again $g_2 = M^{-1}(g_2'; m)$, by reversing the mask operation and using a learnable mask feature vector m.
The features of the first frame g_1 are then passed as *keys* and *values* to the *cross-attention* layers of the decoder, while the features of the second frame g_2 provide the *queries*,

$$\hat{f}_2 = \phi_{\text{dec}}(k = g_1, v = g_1, q = g_2)$$

to reconstruct the unmasked version of the second frame f_2.
As for the loss function the mean squared error between the target frame and its reconstruction $l = \text{MSE}(f_2, \hat{f}_2)$ is used.

4 Data

Following Gupta *et al.* we used Kinetics-400 [7] as our training set. However, due to resource limitations, we restricted our usage to a maximum of 10% of the total dataset. This quantity was sufficient to evaluate the generalization performance of our SiamMAE implementations, while also

ensuring a feasible training duration for this project.
Once the data was downloaded, we filtered the videos to retain only those with a frame rate close to 30 and a minimum duration of 3 seconds. These criteria were necessary to ensure the successful operation of our random temporal sampling, a key aspect of the data loading process.

5 Implementation and Experiments

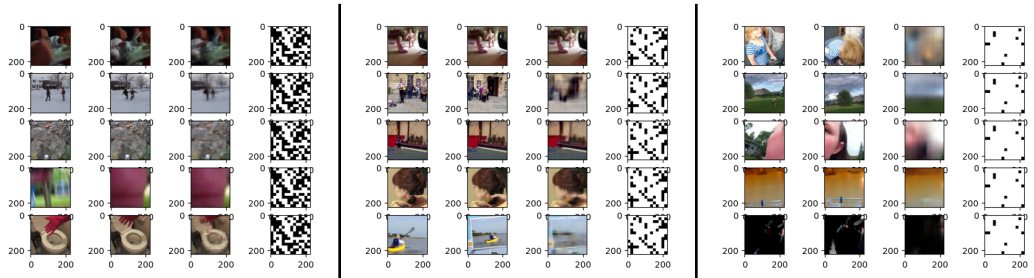
The SiamMAE, similar to Vision Transformers, has a high demand for a huge amount of data and computation resources in order to perform well [3]. As explained in the paper, all their experiments were "performed on 4 Nvidia Titan RTX GPUs for ViT-S/16 models, and on 8 Nvidia Titan RTX GPUs for ViT-S/8 models" [5]. This is beyond the computational resources available for the project, therefore we needed to restrict the number of experiments performed, reduce the size of the dataset heavily and used a smaller architecture, Vit-B/16 rather than Vit-S/16.
We went for two distinct processes:

- We reproduce the paper by developing an entirely independent implementation based solely on the details available in the SiamMAE and cited papers.
- Similar to the paper,"we build on the open-source implementation of MAEs (`https://github.com/facebookresearch/mae`)"[5], which allows us to use their pretrained MAE for the training of the SiamMAE.

5.1 Evaluation on video object segmentation

To evaluate the capabilities of our implementations we measure their performance in the task of video object segmentation. Following SiamMAE we perform video label propagation to determine labels of frames: First, for each of n reference frames $f^{(i)} \in [0,1]^{224\times224\times3}, i \in \{0,\dots,n-1\}$ we calculate the features $g^{(i)} \in \mathbb{R}^{196\times768}$ by passing them through the encoder $g^{(i)} = \phi_{\text{enc}}(f^{(i)})$. We do the same for a target frame $g^{(n)} = \phi_{\text{enc}}(f^{(n)})$, for which we want to determine the label. Next, we calculate the affinity between each feature of the target frame $g_j^{(n)}$ and the features of each reference frame $g_k^{(i)}$ individually $A^{(i)} = g^{(i)} \cdot (g^{(n)})^\top \in \mathbb{R}^{196\times196}$, where $A_{kj}^{(i)} = g_k^{(i)} \cdot (g_j^{(n)})^\top$ represents the affinity between $g_k^{(n)}$ and $g_j^{(n)}$. This affinity matrix encodes the similarity between each combination of feature vectors. To now determine the (patch) label of the target frame $l_n \in [0,1]^{14\times14}$ we use a weighted averaging of the k-nearest neighbors of each feature vector of the target frame.

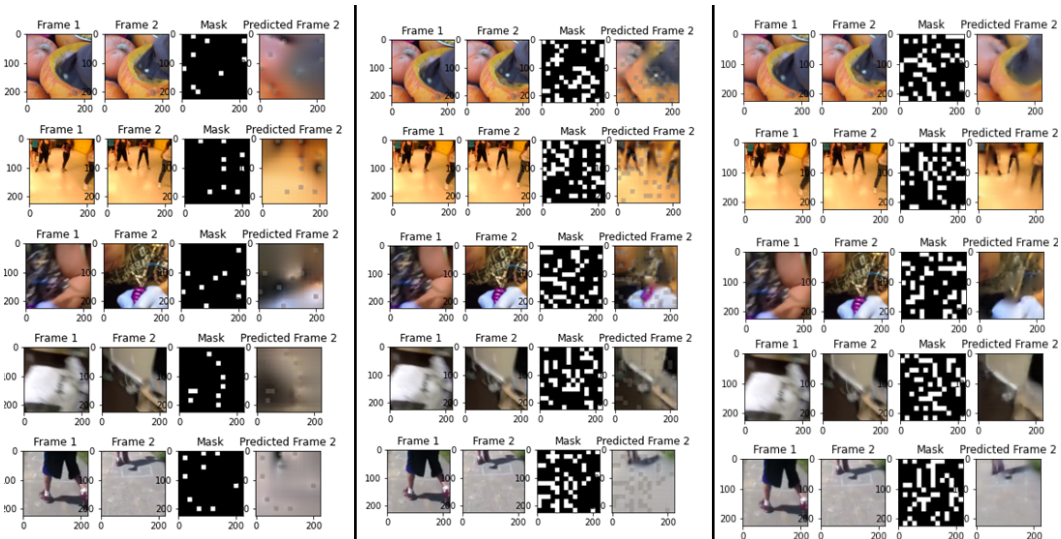
5.2 Implementation from scratch



(a) Mask ratio = 0.5 (b) Mask ratio = 0.75 (c) Mask ratio = 0.95

Figure 2: Qualitative results from the self implemented version during training with three different mask ratios. In each image the first column shows the first frame f_1, the second column the unmasked version of the second frame f_2, which also acts as target. The third column shows the reconstructed output of the model and the fourth column shows the mask (white means that patch is masked).

To develop an implementation of SiamMAE from scratch, i.e. using no code from other projects, demanded a profound comprehension of the architecture and every component thereof. The SiamMAE paper describes the changes relative to the MAE architecture, but assumes familiarity with the latter.



(a) Base with mask ratio = 0.95 (b) Base with mask ratio = 0.75 (c) Large with mask ratio = 0.75

Figure 3: Qualitative results from the MAE version during training with three different mask ratios and the large model. In each image the first column shows the first frame f_1, the second column the unmasked version of the second frame f_2, which also acts as target. The third column shows the mask (white means that patch is masked) and the third column shows the reconstructed output of the model

Consequently, it was imperative to also dive into that paper and subsequently into those of the Vision Transformer and the original Transformer paper.
With this comprehensive knowledge base, we were able to implement all parts required for a SiamMAE architecture, which included elements like Multi-Head-Attention, ViT-Encoder, Cross-Self-Decoder and Masking modules. During the early stages of development, we tested the model's overfitting capacity with limited data sets to validate its learning potential. Upon verification, we transitioned to larger scale training runs.
We used a similar architecture as described for ViT-Base/16 [3] and trained for 20 epochs.
Figure 2 showcases some qualitative results during advanced training stages. It is evident that a lower mask ratio results in superior reconstructions, due to increased availability of information. Remarkably, even at high mask ratios of 0.75 or even 0.95 the model demonstrates the ability to reconstruct the second frame f_2 also for high temporal gaps. This is particularly impressive, considering the model must reconstruct large portions of the frame with sparse information on the new object arrangement.
Although the reconstruction performance of our independently developed version may not achieve the same level as SiamMAE for high mask ratios, our model demonstrates the fundamental ability to learn this task. We attribute the discrepancy to computational limitations. To further support the claim that increased computational resources can mitigate this performance difference, we undertook additional experiments.

5.3 Implementation based on MAE

We further experimented starting from the MAE implementation provided by the authors of MAE [6] and modifying it to fit the SiamMAE architecture, similar to how Gupta *et al.* described in their paper. This approach allowed us to leverage already pretrained models, which significantly reduced the computational requirements to achieve results comparable to those of SiamMAE.
Most experiments were done through the use of the base model, which is a ViT-Base/16 [3] with 12 layers, 768 hidden size, 12 attention heads and 86M parameters. We also experimented with the ViT-Large/16 [3] with 24 layers, 1024 hidden size, 16 attention heads and 307M parameters. We used the same training parameters as in the original paper [5] and trained the model for 10 epochs with 5% of the Kinetics-400 dataset.

Implementation type	mask ratio	$\mathcal{J}_m$	$\mathcal{F}_m$
From scratch	0.5	41.9	56.7
	0.75	41.5	56.2
	0.95	39.1	53.5
Pretrained MAE	0.5	51.7	65.1
	0.75	49.6	63.2
	0.95	52.9	66.3
VideoMAE	-	39.7†	38.9†
DINO	-	60.2†	63.4†
MAE	0.75	52.1†	55.0†
SiamMAE	0.95	60.3†	63.7†

Table 1: Overview of video object segmentation performance on DAVIS-2017 [9]. † indicates values taken from SiamMAE paper [5].

Figure 3 shows the qualitative results of the MAE implementation during training with three different mask ratios (the 0.5 is not shown but it is available in the repository) using the base model and one with the large model. We observe similar results to the self-implemented version, where the reconstruction quality decreases as the mask ratio increases. Furthermore, the reconstructed images have higher quality than the implementation from scratch (as can be seen in Figure 2), with only 10 epochs. Moreover, the results of the large model are very impressive, as the reconstruction quality is very high even for a mask ratio of 0.75.
Unexpectedly, the results of the MAE implementation were almost as good as the paper for the base model and even beat the paper for $\mathcal{F}_m$ in multiple masking rations (as can be seen in Table 1). Furthermore, we found that the model trained with 0.95 mask ratio performed the best across all metrics. We speculate that this is due to the fact that the model is forced to learn the most information from the first frame, which is the most important frame for the video object segmentation task. However, we were not able to beat $\mathcal{J}_m$ where we performed worse than the paper for all masking ratios.
As expected, the pretrained MAE variant of SiamMAE performed better than the from scratch even with less epochs and data. This is likely because the original MAE was pretrained on a very large dataset, therefore we were able to do transfer learning and receive very good results. This also means that our self-implemented SiamMAE would have replicated the original SiamMAE given more data and computation resources.

5.4 Challenges

One main challenge in both approaches was the handling of the data. Some of the downloaded videos were corrupted, therefore we needed to filter them with a script. For the dataloading we had to use pytorchvideo [4] instead of an existing PyTorch dataloader, and implement random temporal frame sampling.
Within the training phase, we pinpointed data loading as a constraint, primarily due to the extensive overhead of video decoding and the slight amount of data we employed from each video. This lead to a very low utilization of the GPU, and therefore extremely long training routines. In an effort to alleviate this constraint, we preprocessed the data by transforming video files into directories of frame files. Simultaneously, we also adjusted their scale as needed to facilitate loading with minimal overhead. This lead together with multiple, parallel dataloaders to a 25-fold increase in processing speed compared to the previous, non-optimized method.

5.4.1 From Scratch

Developing the whole architecture from scratch is a complex task, since the SiamMAE architecture is based on a variety of modules which interact with each other. A small incorrectness in some part may not be directly visible but lead to an inferior learning capability of some other part. A major problem were horizontal stripe artifacts in the reconstructed output, which did not vanish even in an artificial overfitting setting (single sample training). After using the two-dimensional positional embedding from the official MAE implementation, those artifacts were removed.

5.4.2 From MAE

When implementing the paper, starting from the MAE [6], we faced multiple challenges. The primary challenge was understanding SiamMAE architecture since it was only briefly described in the paper and the authors did not publish their code. As a result, we needed to first analyse and understand exactly how MAE worked before we could even begin work with the SiamMAE. The next challenge was the heavy computational demands. Training the SiamMAE took a huge amount of time, which meant that every experiment was performed over multiple days and if there were any implementation issues, this would require multiple days to fix and re-run. In addition, we needed to ensure the hyperparameters were exactly the same as the SiamMAE [5] papers, otherwise we received terrible results.

6 Conclusion

The SiamMAE paper [5] was difficult to reproduce due to the requirement for a huge amount of data and computation. We were able to replicate the paper with two distinct approaches: a self-implemented variant and from-MAE variant. Both approaches performed well but the use of the from-MAE had a higher performance since then we could use a pretrained MAE that was trained with a lot of images. The main problem with the paper is that they only briefly describe SiamMAE and leave a lot up to interpretation and referring to other papers. Nonetheless, we receive similar results to the paper in the task of video object segmentation and our qualitative results were also good. In addition, we expanded on the paper by experimenting with other masking ratios for the siamese encoder and cross-self decoder architecture.

6.1 Ethical Consideration, Societal Impact, Alignment with UN SDG targets

The SiamMAE paper [5] does not discuss any ethical concerns, social impacts or alignment with UN SDG targets. However we believe, the societal impact of the paper is positive, as it provides a new method to improve the performance of video object segmentation and other tasks. We also think this contributes to the UN SDG Target 9.5, which aims to enhance scientific research and upgrade the technological capabilities of industrial sectors in all countries. Our paper pushes the envelope further by publicising our SiamMAE implementation, allowing others to further develop it.

6.2 Code Repository

We have implemented in two different repositories the code for the reproduction of SiamMAE. The first one is a self implemented version of SiamMAE, which can be found at `https://github.com/aweers/replicate_siamMAE`. The second one is a version based on the MAE implementation, which can be found at `https://github.com/eleander/SiamMAE`.
Furthermore, we have made the repositories public and documented them, as there are currently no public implementations of SiamMAE. We hope that this will help other researchers to reproduce and build upon the work of Gupta *et al.* [5].

7 Self Assessment

We believe that our project is deserving of Excellent (A) for many reasons:

- **Results Match Paper**: Both of our implementations receive very nice qualitative and quantitative results that match the paper for a masking ratio of 50%, 75% and 95% using only the ViT-B/16 model. When we used the pretrained MAE large then we received comparable results to the paper for the reconstruction.
- **Extremely Heavy Computation Demands**: The SiamMAE paper has insanely high computational demands and data demands. The original paper required 8 GPUs in order to receive state-of-the-art performance. We only had used 2 GPUs, a RTX Nvidia 3070 and a RTX Nvidia 4090, which both has a lower VRAM than their GPU (e.g. we couldn't run bigger models) and is much slower than 8 GPUs. As a result, we needed to restrict the number of experiments performed and use smaller architectures.

- **More Experiments**: We added to the original paper by experimenting with different masking ratios on the siamese encoder and self-cross decoder setup. We focused on video object segmentation due to the computational constraints.
- **Released Implementation as Public Repository**: The original paper does not release their code and only briefly mentions that they build on the open-source implementation of MAE. Therefore, our publicised repository allows scientists to see how SiamMAE could be implemented and fork off of it to expand upon the experiments. We also included extensive documentation in our code so that researchers can instantly understand what we did.

7.1 Self-nomination for Bonus (optional)

We would like to self-nominate ourselves for a bonus in the following sections:

- **Successful reimplementation in a deep learning framework for which an online public repository is not available**: SiamMAE did not have a public repository, therefore we needed to study multiple paper and figure out how to implement it solo.
- **Difficulty of the implementation: we understand that some papers might be significantly more difficult than others to implement**: Reimplementing SiamMAE requires an understanding of MAE, ViT, Transformers and video label propagation before it is even possible to implement, therefore significant amount of time was spent understanding the different papers.

References

[1] Jane Bromley et al. "Signature Verification Using a "Siamese" Time Delay Neural Network". In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1993, pp. 737–744.

[2] Mathilde Caron et al. "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 9650–9660.

[3] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR* (2021).

[4] Haoqi Fan et al. "PyTorchVideo: A Deep Learning Library for Video Understanding". In: *Proceedings of the 29th ACM International Conference on Multimedia*. `https://pytorchvideo.org/`. 2021.

[5] Agrim Gupta et al. "Siamese Masked Autoencoders". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: `https://openreview.net/forum?id=yC3q7vInux`.

[6] Kaiming He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 15979–15988. DOI: `10.1109/CVPR52688.2022.01553`.

[7] Will Kay et al. *The Kinetics Human Action Video Dataset*. 2017. arXiv: `1705.06950 [cs.CV]`.

[8] Daniel McKee et al. *Transfer of Representations to Video Label Propagation: Implementation Factors Matter*. 2022. arXiv: `2203.05553 [cs.CV]`.

[9] Jordi Pont-Tuset et al. "The 2017 DAVIS Challenge on Video Object Segmentation". In: *arXiv:1704.00675* (2017).

[10] Zhan Tong et al. "VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training". In: *Advances in Neural Information Processing Systems*. 2022.

[11] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.