A APPENDIX

A.1 IMPLEMENTATION DETAILS

In Section 3, we explained that by saving in memory the transformer results, we were able to go from a quadratic complexity to a linear one. We give more details about this in Figure 6. In the memory-based version we forward only one frame, the last one, through the model. Only this frame has to be inpainted in the transformers, but we can still use information from the other frames, stored in the memory. Therefore, the query represents this one frame while the keys and values represent all the frames of the input (frame to inpaint + neighboring frames + reference frames).

We also add Figure 7 to explain where the 'memory operations' are done in the classical transformer pipeline.

A.2 PARAMETERS TUNING

We present here the influence of some parameters of the model on the performances. The model under scrutiny is the refined memory-based model with the FuseFormer backbone, as it is one of the most promising.

In addition to the actual frame to inpaint, this model's input is composed of three parts, as already shown in Equation equation 4 and Figure 3:

1. Neighboring frames from the online inpainter's memory
2. Neighboring frames from the refining inpainter's memory
3. Reference frames from the refining inpainter's memory

Note that because we want to work on live videos that will possibly be very long, it would become impossible to take all the previous reference frames, regardless of the choice of the sampling rate. Therefore, only the last n reference frames are taken in our models, with n a tuning parameter.

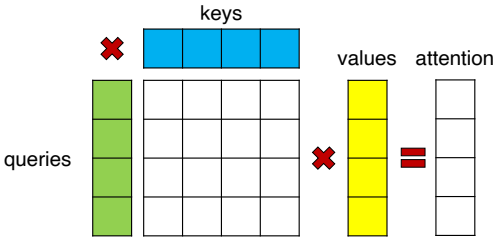
In Tables 4 and 5, we propose a study of the performance of the model when tuning the size of these different parts. As we could expect, increasing the size of the input makes the model slower. However, the added frames do not have the same impact on the inpainting performance. In particular, we can observe that it is preferable to add more reference frames from the refined memory than neighboring frames. This confirms the paramount importance of the reference frames that was already highlighted in the ablation study in Table 3.

For all types of frames, adding more of them for the input increases the quality up to a certain point where more frames don't increase the PSNR anymore, or not by much. Combined with the progressive loss of FPS, this encourages us to keep the size of the input reasonable. For all the numerical results given in this work, and especially for Tables 1 and 2, we took for each category a number of 3 frames to try to get good quality without leaving the 20 FPS domain.

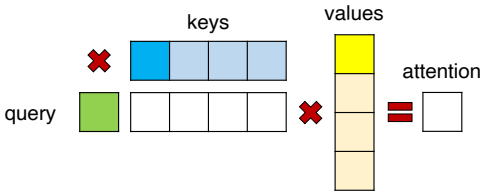
A.3 MORE DETAILS ON THE LIMITATIONS AND POSSIBLE WORKAROUNDS

While giving promising results in online inpainting, our models still show some limitations that we want to address here. Some of them are specific to our work, giving a path for prospective improvements in the future. This includes the gap in quality that still remains after increasing the frame rate, as it may make our models less desirable than the current ones, slower but with higher quality. Another obvious limitation is the fact that our framework is, as of now, only adapted to transformer-based inpainting algorithms. Nevertheless, we think that our idea to save intermediate results from the former frames to help the inpainting of the new ones can work with other types of inpainting, for example the flow-based ones.

Other limitations, however, seem to be more inherent to the online inpainting problem we are trying to solve. For example, due to the lack of numerous supporting frames at the beginning, the first inpainted frames are significantly poorer, before seeing an increase in quality later. This problem, also discussed in A.5, is inseparable from the context of online inpainting: at the very beginning



(a) Without memory



(b) With memory

Figure 6: Attention calculations with and without the memory. **(a)** Without memory, every frame has to be inpainted by each transformer. The queries represent the n frames of the input, as well as the keys and the values. A normal attention calculation is performed, which has quadratic complexity. **(b)** When using the memory, we only need to calculate the inpainting for the new frame, represented in the query. The keys and values stay the same as the context surrounding this frame. The operation is linear this time.

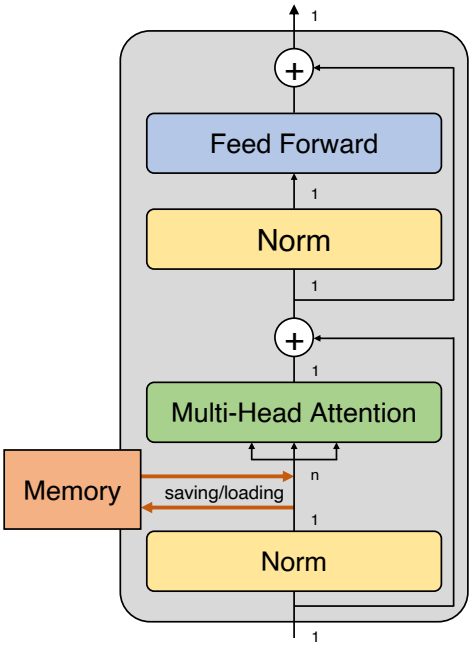


Figure 7: Saving operation in the transformer. Our transformers follow the same pipeline as classical vision transformers. The only difference is the memory saving and loading (also shown in Figure 2) happening just before the attention calculation. We also indicate the number of frames forwarded at each step: only one frame is kept through the whole model and the contextual frames are only present when needed.

Table 4: Parameter tuning for the refined memory

Neighboring frames	Reference frames	PSNR	FPS
1	1	30.46	24.5
1	2	30.59	23.5
1	3	30.69	22.2
1	4	30.75	21.5
1	5	30.77	21.1
1	6	30.78	20.8
2	1	30.52	22.7
2	2	30.62	21.7
2	3	30.71	21.7
2	4	30.79	21.0
2	5	30.82	20.7
2	6	30.82	20.6
3	1	30.54	22.8
3	2	30.66	21.8
3	3	30.76	20.5
3	4	30.82	20.4
3	5	30.85	20.0
3	6	30.86	19.9
4	1	30.57	22.3
4	2	30.69	21.4
4	3	30.79	20.1
4	4	30.84	20.1
4	5	30.88	19.7
4	6	30.87	19.3
5	1	30.57	22.3
5	2	30.69	21.2
5	3	30.78	20.6
5	4	30.86	19.7
5	5	30.87	19.2
5	6	30.90	18.9
6	1	30.57	22.2
6	2	30.68	21.4
6	3	30.77	20.6
6	4	30.85	19.7
6	5	30.86	19.4
6	6	30.90	18.8

Table 5: Parameter tuning for the online memory

Neighboring frames	PSNR	FPS
1	30.73	22.0
2	30.83	20.4
3	30.87	19.2
4	30.92	18.0
5	30.91	17.5
6	30.91	16.7

Table 6: Memory usage of the FuseFormer-based models on both datasets (in GB).

Model	DAVIS	YouTube-VOS
Online	0.53	0.73
+ Memory	1.28	2.26
+ Refined	1.43	2.57

of a live video, we almost have no information available. A possible workaround could be to try to completely hallucinate the masked content using an image inpainting method, rather than leveraging supporting frames that do not exist yet in the video. In any case, we do not think it is a truly crippling problem, as we can always start the inpainting after a small delay to accumulate enough information for the first inpainted frames.

Another intrinsic limit of online video inpainting appears when an object is at first hidden by the mask, before being revealed later in the video. Indeed, it may be impossible to guess this hidden content in advance, creating an obvious discontinuity when the object finally leaves the mask to appear in the rest of the video. This may become even worse if the mask lies on the border of the video, as new objects can enter the frame while hidden by the mask, making predictions of what is under the mask simply impossible. This, of course, is not a problem on a bounded video with offline inpainting because the model uses reference frames in the whole video. Therefore, it can predict content in advance by simply seeing it appear in the video in future frames. To this day, we do not see solutions to address this issue in the case of online inpainting, as we purely cannot see in the future.

A.4 Memory Usage

Storing intermediate results in memory saves a lot of time but it requires more memory in return. To evaluate our work from this perspective, we measured the GPUs memory usage for our three online models on the FuseFormer, on both DAVIS and YouTube-VOS datasets. The results are given in Table 6. This includes the memory used to have the inpainting model loaded on the GPU, but this part is minority. The biggest part of the memory is taken by the tensors (original masks and frames or inpainting memory) used as input by the model. For the refined model, the value given corresponds to the sum of memory usage on the two GPUs.

We can observe a clear increase in the memory usage for the models storing inpainting for the later frames. Even thought the difference is less significant for both memory-based models, more memory is required for the refined model as it stores more information and requires a second model.

The difference of GPU usage between both datasets can be explained by the average length of the corresponding videos. YouTube-VOS videos tend to be longer than the DAVIS ones, creating bigger memory tensors. This was not a problem in this project as our GPUs had an higher memory limit (12 GB), but this would raise memory issues when dealing with longer videos as we would hit that limit. In that case, it would become necessary to dynamically maintain the inpainting memory by removing information from ancient frames to store the new ones. It is reasonable to affirm that this wouldn't affect the general quality of the results as we never use too old frames to inpaint the new ones anyway. However, we cannot discard the possibility that this could slow down the process a bit.

A.5 Temporal Analysis

Because online models can also use information from the past, one can expect them to perform poorly at the beginning of a video and then to improve with more and more information available. In Figure 8, we calculated the mean PSNR and SSIM at each frame for the videos of the YouTube-VOS dataset. The models here use the FuseFormer backbone and are compared to the offline approach that can use information from everywhere in the videos. For both the online and the refined memory-based models, we can clearly note the gap of quality decreasing as we progress in the video, up to a

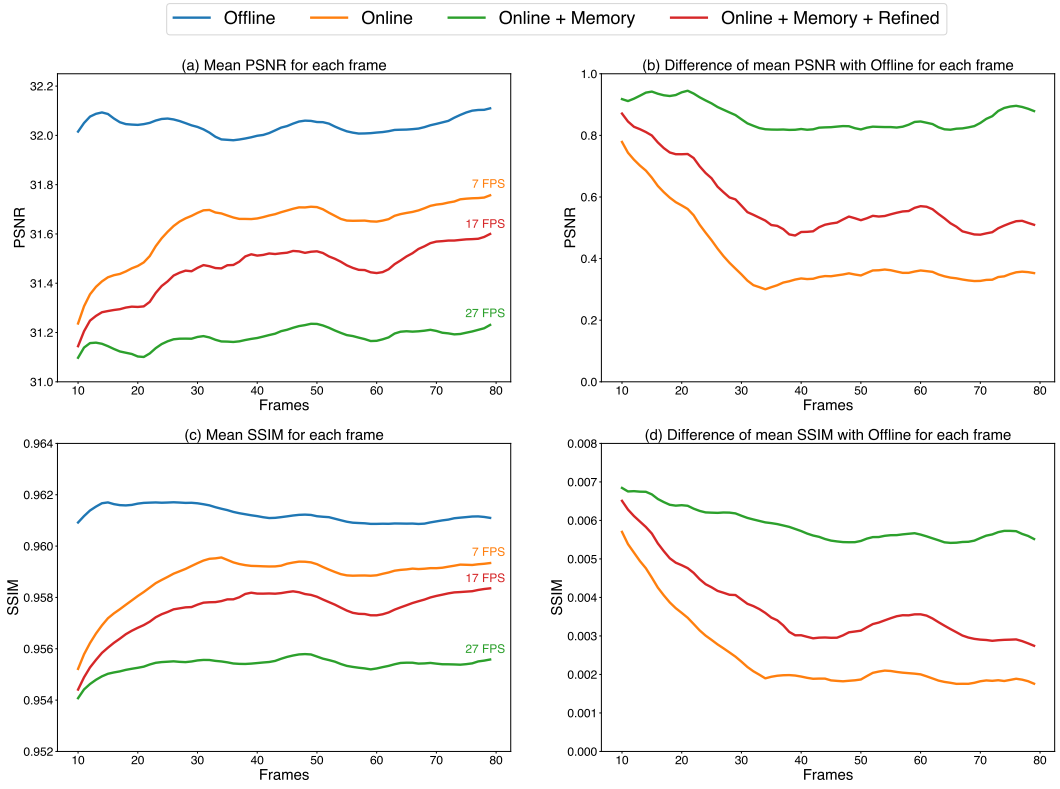


Figure 8: Mean PSNR and SSIM at each frame on YouTube-VOS (500+ videos). Models use FuseFormer backbone, values are smoothed with a moving average of 10 frames. On the right, we show the values differences with the offline model. The two most performing models are able to partly close the quality gap with the offline one as they discover more frames to use.

certain point. This shows that these inpainters can leverage the new information made available as the video continues.

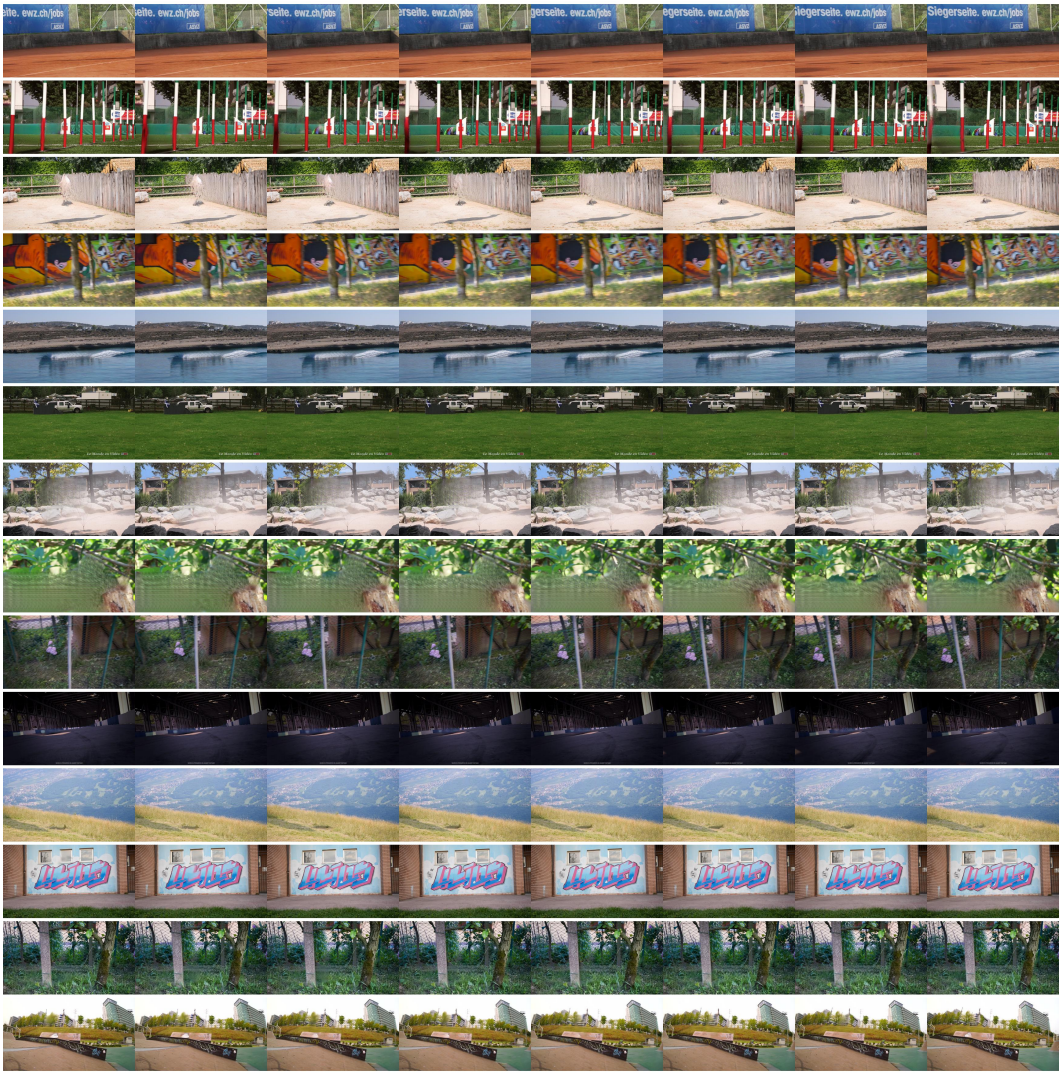


Figure 9: Visual results of the object removal task. The frames are shown consecutively. See the link in Section A.6 for full videos.

A.6 Additional Visuals

In Section 4.6, we selected four videos from the DAVIS dataset to show visual results of the object removal task using the refined memory-based model. We have chosen these videos in particular because they also appear in previous publications. On top of that, we randomly selected 10 more videos out of the 90 videos available in DAVIS dataset. In Figure 9, we show consecutive inpaited frames for these 14 videos. Full videos compared side by side with the original videos and the classic online inpaintings are anonymously available at: `https://github.com/ICLR-23-Supp/Supplementary-Material`.