Self-supervised Amodal Video Object Segmentation

Jian Yao[1]* **Yuxin Hong**[2*] **Chiyu Wang**[3*] **Tianjun Xiao**[4†] **Tong He**[4]
Francesco Locatello[4] **David Wipf**[4] **Yanwei Fu**[2†] **Zheng Zhang**[4]
[1] School of Management, Fudan University
[2] School of Data Science, Fudan University
[3] University of California, Berkeley
[4] Amazon Web Services
{jianyao20, yxhong20, yanweifu}@fudan.edu.cn, wcy_james@berkeley.edu
{tianjux, htong, locatelf, daviwipf, zhaz}@amazon.com

Abstract

Amodal perception requires inferring the full shape of an object that is partially occluded. This task is particularly challenging on two levels: (1) it requires more information than what is contained in the instant retina or imaging sensor, (2) it is difficult to obtain enough well-annotated amodal labels for supervision. To this end, this paper develops a new framework of Self-supervised amodal Video object segmentation (SaVos). Our method efficiently leverages the visual information of video temporal sequences to infer the amodal mask of objects. The key intuition is that the occluded part of an object can be explained away if that part is visible in other frames, possibly deformed as long as the deformation can be reasonably learned. Accordingly, we derive a novel self-supervised learning paradigm that efficiently utilizes the visible object parts as the supervision to guide the training on videos. In addition to learning type prior to complete masks for known types, SaVos also learns the spatiotemporal prior, which is also useful for the amodal task and could generalize to unseen types. The proposed framework achieves the state-of-the-art performance on the synthetic amodal segmentation benchmark FISHBOWL and the real world benchmark KINS-Video-Car. Further, it lends itself well to being transferred to novel distributions using test-time adaptation, outperforming existing models even after the transfer to a new distribution.

1 Introduction

Cognitive scientists have found human vision system contains several hierarchies. Visual perception [27] first carves a scene at its physical joints, decomposing it into initial object representation by grouping and simple completion. At this point, the representation is tethered into the retina sensor [13]. Then, correspondence or motion on the temporal dimension is built to form the object representation that is untethered from the retinal reference frame through operations like spatiotemporal aggregation, tracking, inference and prediction [6]. The more stable untethered representation is ready to be raised from the perception system to the cognitive system for higher level action and symbolic cognition [23]. Machine learning, especially with artificial neural networks, has progressed tremendously on tethered vision tasks like detection and modal segmentation. The natural next step is to go the higher rung of the ladder by tackling untethered vision.

This paper studies the task of amodal segmentation which aims at inferring the whole shape of the object on both visible and occluded parts. It has critical applications on robot manipulation and

*Work completed during internship at AWS Shanghai AI Labs.
†Correspondence authors are Tianjun Xiao and Yanwei Fu.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

autonomous driving [24]. Conceptually, this task is on the bridge between tethered and untethered representations. Amodal segmentation requires prior knowledge. One option that has been explored in literature is using the tethered representation and prior knowledge about object type to get amodal mask. Alternatively, we can get amodal masks using the untethered representation by building dense object motion across frames to explain away occlusion, which is referred as spatiotemporal prior. We prefer to explore more on the second one since the dependence on type prior makes the first method hard to generalize, considering the frequency distribution of visual categories in daily life is long-tailed.

Following this direction, we propose a Self-supervised amodal Video object segmentation (SaVos) pipeline which simultaneously models amodal mask and the dense object motion on the amodal mask. Unlike traditional optical flow or correspondence networks, our approach does not require explicit visual correspondence across pixels, which would be impossible due to occlusions. Instead, modeling motion using temporal information allows us to complete dense amodal motion predictions.

The architecture is built for spatiotemporal modeling, which has better generalization performance than using type priors. Despite that, we show that SaVos automatically figures its way to learn type prior as well, as learning types can help the encoder-decoder-style architecture make prediction. This makes generalization to distribution shifts remain challenging, for example, to unseen types of objects. To address this issue, we need to suppress the type prior and amplify spatiotemporal prior to make predictions. This is achieved by combining SaVos with test-time adaptation. Critically, we found that our model is "adaptation-friendly" as it can naturally be improved with test-time adaptation techniques without any change on the self-supervised loss, achieving a significant boost in generalization performance.

We make several contributions in this paper:

(1) We propose a Self-supervised amodal Video object segmentation (SaVos) training pipeline built upon the intuition that the occluded part of an object can be explained away if that part is visible in other frames (Figure 1), possibly deformed as long as the deformation can be reasonably learned. The pipeline turns visible masks in other frames to amodal self-supervision signals.

(2) The proposed approach simultaneously models the amodal mask and the dense amodal object motion. The dense amodal object motion builds the bridge between different frames to achieve the transition from visible masks to amodal supervision. To address the challenge of predicting motion on the occluded area, we propose a novel architecture design that takes the advantage of the inductive bias from the spatiotemporal modeling and the common-fate principle of Gestalt Psychology [39]. The proposed method shows the state-of-the-art amodal segmentation performance in self-supervised setting on several simulation and real-world benchmarks.

(3) The proposed SaVos model shows strong generalization performance on drastic distribution shifts between training and test data after combining with one-shot test-time adaptation. We empirically demonstrate that, by applying test-time adaptation without any change on the loss, SaVos trained on synthetic fish dataset can even outperform a competitor that is well learned on the target real-world driving car dataset. Interestingly, applying test-time adaptation on an image-level baseline model doesn't bring the same improvement as observed on SaVos. This provides an unique perspective on comparing different models by checking how effective can test-time adaptation work on them.

2 Related works

Untethered vision and amodal segmentation. Human vision forms a hierarchy by grouping retina signals into initial object concept; and the representation will untether from the immediate retina sensor input grouping the spatiotemporally disjoint pieces. Such untethered representation has been studied in various topics [23, 21, 28, 26, 34, 22]. Particularly, Amodal segmentation [46] is a task inferring shape of the object on both visible and occluded part. There are various image amodal datasets such as COCOA [46] and KINS [24], and video amodal dataset – SAIL-VOS [11] created by the GTA game engine. Unfortunately, SAIL-VOS has frequent camera view switches, not the ideal testbed to apply video tracking or motion. Several efforts are made towards amodal segmentation on these datasets [46, 24, 7, 45, 41, 44, 33, 17, 43, 30, 20]. Generally speaking, most of the methods are on image level and they model type priors with shape statistics, as such it is challenging to extend their models to open-world applications where object category distributions are long-tail. Amodal segmentation is also related to structured generative model [47, 16, 29, 18]. These models attempt to

maximize the likelihood of the whole video sequences during training so as to learn a more consistent object representation and the amodal representation. However, the major tasks for those models are object discovery and presentation and they are tested on simpler datasets; self-supervised object discovery in real-world complex scenes like the driving scene in [9] remains too challenging for these methods. Without object discovered, no proper amodal prediction can be expected.

Dense correspondence and motion Our goal is to achieve amodal using untethered process, which requires object motion signals. There have been studies [8, 2] on correspondence and motion before deep learning time. FlowNet [5] and its follow-up work FlowNet2 [14] train deep networks in a supervised way using simulation videos. Truong et al.[34] proposes GLU-Net, a global-local universal network for dense correspondences. However, motion on the occlusion area cannot be estimated with those methods. Occlusion and correspondence estimation depend on each other and it is a typical chicken-and-egg problem [15]. We need to model additional priors.

Video inpainting A related but different task is video inpainting. Existing video inpainting methods fill the spatio-temporal holes by encourage the spatial and temporal coherence and smoothness [40, 10, 12], rather than particularly inferring the occluded objects. The object-level knowledge was not explicitly leveraged to inform the model learning. Recently, Ke et al.[17] learns object completion by contributing the large-scale dataset Youtube-VOI, where occlusion masks are generated using high-fidelity simulation to provide training signal. Nevertheless, there is still the *reality gap* between synthetic occlusions and the amodal masks in the real-world . Accordingly, our model is designed to learn the amodal supervision signal in the easily accessible raw videos if spatiotemporal information is properly mined.

Domain generalization and test-time adaptation Transfer learning [4] and domain adaptation [25] are general approaches for improving the performance of predictive models when training and test data come from different distributions. Sun et al.[31] proposes Test-time Training. It is different from finetuning where some labeled data are available in test domain, and different from domain adaptation where there is access to both train and test samples. They design a self-supervised loss to train together with the supervised loss and in test time, apply the self-supervised loss on the test sample. Wang et al.[37] proposes a fully test time adaptation by test entropy minimization. A related topic is Deep Image Prior [36] and Deep Video Prior [19], they directly optimize on test samples without training on training set. Our model is self-supervised thus naturally fit into test-time adaptation framework. We will see how it works for our method on challenging adaptation scenarios.

3 Method

Notations . Given the input video $\{\mathbf{I}_t\}_{t=1}^{T}$ of T frames with K objects, the task is to generate the amodal "binary" segmentation mask sequences $\mathbf{M} = \left\{M_t^k\right\}$ for each object in every frame. On the raw frames, we obtain the image patch I_t^k and visible modal segmentation mask $\mathbf{V} = \left\{V_t^k\right\}$. Further, we also obtain the optical flow $\Delta V_t = \left\{\Delta V_{\mathrm{x},t}^k, \Delta V_{\mathrm{y},t}^k\right\}$ such that $I_{t+1}[x + \Delta V_{\mathrm{x},t}[x, y], y + \Delta V_{\mathrm{y},t}[x, y]] \approx I_t[x, y]$. These information can be retrieved from human annotation or extracted with off-the-shelf models, and we use them as given input to our model.

3.1 Overview of Our SaVos Learning Problem

The key insight of the amodal segmentation task is to maximally exploit and explore visual prior patterns to *explain away* the occluded object parts [35]. Such prior patterns include, but are not be limited to (1) **type prior**: the statistics of images, or the shape of certain types of objects; and (2) **spatiotemporal prior**: the current-occluded part of an object might be visible in the other frames, as illustrated in Figure 1. Under a self-supervised setting, we exploit temporal correlation among frames relying exclusively on the data itself.

Specifically, SaVos generates training supervision signals by investigating the relations between the amodal masks and visible masks on neighboring frames. The key assumption is "some part is occluded at some time, but not all parts all the time", and that deformation of the past visible parts can be approximately learned. I.e. gleaning visible parts over enough frames produces enough evidences to complete an object. The inductive bias we can and must leverage is spatiotemporal continuity. We note that parts remain occluded all the time can not be recovered unless there are other priors

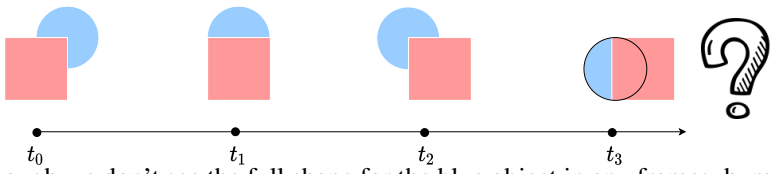

Figure 1: Though we don't see the full shape for the blue object in any frames, human can tell the full shape of that object at frame t_3. We just need to stitch seen parts in different frames.

such as classes and types, which is also learnable in our model design. Figure 2 a) shows the training pipeline.

Prediction of amodal mask and amodal motion At frame t for object k, SaVos predicts the amodal mask $\tilde{M}_t^k$ with

$$\tilde{M}_t^k, \Delta\tilde{M}_t^k = \mathrm{F}_\theta(I_{\leq t}^k, V_{\leq t}^k, \Delta V_{\leq t}^k) \tag{1}$$

where F_θ is a learnable module paramterized by θ with more detailed introduction in Section 3.3, and $\Delta\tilde{M}_t^k$ is the dense motion on the amodal mask. Ideally F_θ is able to aggregate available visual, semantic and motion information from all the currently available frames. The intuition here is that seeing various parts of an object (i.e. $\mathbf{V}$) and their motions ($\mathbf{\Delta V}$) is enough to reconstruct $\mathbf{M}$. However, leaving as is, there can be an infinite number of explain-away solutions.

The first obvious training signal is to check the consistency between $\tilde{M}_t^k$ and V_t^k. The signal is weak since the model can learn a simple copy function. A stronger signal is to assume a generative component to predict the amodal mask in the *next* time frame and use visible V_{t+1}^k in addition. Since transformation is already predicted by $\Delta\tilde{M}_t^k$, we can obtain an estimation of the amodal mask at frame $t+1$ by a warping function, i.e. $\hat{M}_{t+1}^k = \mathrm{Warp}(\tilde{M}_t^k, \Delta\tilde{M}_t^k)$. $\hat{M}_{t+1}^k$ satisfies

$$\hat{M}_{t+1}^k[x + \Delta\tilde{M}_{\mathrm{x},t}^k[x,y], y + \Delta\tilde{M}_{\mathrm{y},t}^k[x,y]] \equiv \tilde{M}_t^k[x,y]. \tag{2}$$

Now we compute the distance between $\hat{M}_{t+1}^k$ and V_{t+1}^k with the assumption that V_{t+1}^k might includes parts occluded in V_t^k, and define the first training loss as

$$\mathcal{L}_M = \sum_{k=1}^{K}\sum_{t=1}^{T-1} d_M\left(\hat{M}_{t+1}^k, V_{t+1}^k\right) \tag{3}$$

where

$$d_M\left(\hat{M}_{t+1}^k, V_{t+1}^k\right) = W_{t+1}^k \odot \mathrm{BCE}\left(\hat{M}_{t+1}^k, V_{t+1}^k\right) \tag{4}$$

with BCE being the vanilla form of binary cross entropy loss function, and W_{t+1}^k being a weight matrix that masks out the area occluded at $t+1$ for object k. Concretely,

$$W_{t+1}^k = \left(\mathbf{1} - \sum_{i=1}^{K} V_{t+1}^i\right) + V_{t+1}^k \tag{5}$$

where $\mathbf{1}$ is a all-one matrix with the same shape as the mask tensor. With this mask, $\mathcal{L}_M$ only generates supervision signal on the visible parts and background between different frames, with the occluded part being masked out from providing any feedback. Our prediction can be *under*-complete if only $\mathcal{L}_M$ is applied, since all that the model is forced to learn is the visible masks one frame later.

Amodal consistency loss The temporal consistency loss $\mathcal{L}_C$, assuming some distance measure function $d(\cdot,\cdot)$ is straightforward in its form:

$$\mathcal{L}_C = \sum_{k=1}^{K}\sum_{t=2}^{T} d_C\left(\tilde{M}_t^k, \hat{M}_t^k\right) \tag{6}$$

The intuition is that the amodal mask prediction $\tilde{M}_t^k$ at t should be consistent with the estimation $\hat{M}_t^k$ warped from $t-1$. $\tilde{M}_t^k$ includes new evidence $(V_t^k, \Delta V_t^k)$ that is not available when computing $\hat{M}_t^k$, forcing the generative component in estimating $\hat{M}_t^k$ to catch up. This loss links the supervision signals in all frames to guide the prediction in each frame.

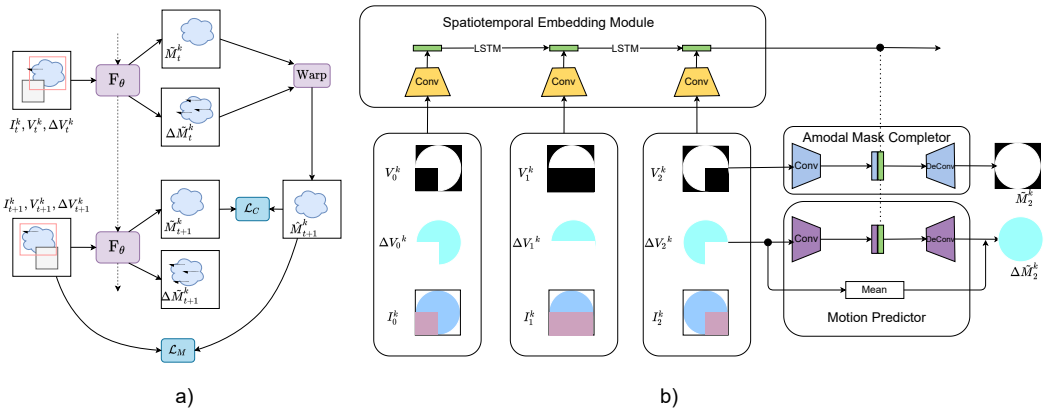

Figure 2: a) SaVos Training Pipeline. b) SaVos Architecture

Consider an object that is made up by two parts, each is visible in only one of the two adjacent frames, a good estimation is encouraged to include both parts under this loss. In this work, we use Diff_IoU as the metric d_C, introduced in [3], since it's symmetric to inputs. Note that this loss has a similar form of temporal cycle-consistency [38].

Combining Equation 4 and Equation 6, the final loss is:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_M + \lambda_2 \cdot \mathcal{L}_C \tag{7}$$

Analysis on necessity and sufficiency of $\mathcal{L}$ Define $\mathcal{L}_M^k = \sum_{t=1}^{T-1} d_M\left(\hat{M}_{t+1}^k, V_{t+1}^k\right)$ and $\mathcal{L}_C^k = \sum_{t=2}^{T} d_C\left(\tilde{M}_t^k, \hat{M}_t^k\right)$. It is easy to show that $\mathcal{L}_M^k = \mathcal{L}_C^k = 0$ is necessary for $\tilde{M}_t^k = \hat{M}_t^k = M_t^k$. Specifically, for any t and k, we have $d_M\left(\hat{M}_t^k, V_t^k\right) = d_C\left(\tilde{M}_t^k, \hat{M}_t^k\right) = 0$ if $\tilde{M}_t^k = \hat{M}_t^k = M_t^k$, and that directly leads to $\mathcal{L}_M^k = \mathcal{L}_C^k = 0$. We further analyzes the sufficiency of $\mathcal{L}_M^k = \mathcal{L}_C^k = 0$. The theorem statement and its proof can be found in supplementary material.

Some pixels of an object might never be visible in any frames. For example, parked cars that line up along the roadside with corners invisible, e.g. the target car in Figure 3. We emphasize that our model can still align the correct amodal segmentation with the global optima of the proposed loss. Consequently, on those cases, our method can still work at least as good as image-level methods since it is also able to capture type prior for known types with the architecture design in Section 3.3. The encoder-decoder architecture contains an information bottleneck, which makes it easy to squeeze out type information since it's concise and beneficial to amodal prediction. In addition, the fewer pixels remain invisible all the way, the more spatiotemporal information our model can leverage to improve its performance.

Bi-directional Prediction SaVos as described so far suffers from cold start problem, i.e. the first few frames may not be informative enough. We solve this by simply predicting backwards in time. To merge the prediction from both directions, we add an alpha channel prediction together with the amodal mask and let the model decide which side to trust more:

$$\hat{M}_t^k = \overrightarrow{\hat{\alpha}_t^k} \odot \overrightarrow{\hat{M}_t^k} + \overleftarrow{\hat{\alpha}_t^k} \odot \overleftarrow{\hat{M}_t^k} \tag{8}$$

where $\overrightarrow{\hat{\alpha}_t^k}$ and $\overrightarrow{\hat{\alpha}_t^k}$ are the alpha channel from each direction normalized with each other using Softmax function. $\overrightarrow{\hat{M}_t^k}$ and $\overleftarrow{\hat{M}_t^k}$ are the predicted amodal mask from each direction.

3.2 Test-time Adaptation for SaVos

SaVos models spatiotemporal prior and as such should be robust against data distribution shift. However, certain components (especially the generative part) are sensitive to samples in the training data. Type prior can be implicitly learned during training, which the model falls back on and struggles to "synthesize" novel masks. In this work we adopt one-shot test-time adaptation as stated in [32]. We don't expect new knowledge to be learned on single sample, but to suppress unnecessary type

prior and rely only on learned spatiotemopral prior. The advantage is that a base model can be reused to tackle new data distribution which is not expected to be part of the long term sample repository.

Since training is on each single test sample independently, no change for the testing environment is required, only the inference time will increase. In [32], a test-time self-supervised loss is used to help the model adapt to the test data distribution. Theoretically, if the gradient of the test-time loss is on the same direction of the main training-time loss, the model performance on the test domain will be improved. As our SaVos model is self-supervisedly learned, the test-time adaptation naturally apply. In practice, we optimize the loss on Equation 7 on a test video *without any change*.

In experiments, test-time adaptation indeed helps on challenging distribution shifts test scenarios. We will have more detailed analysis on the learning dynamics and efficiency on the test-time adaptation for SaVos in the Experiment Section 4 comparing with the baseline.

3.3 Architecture

As depicted in Figure 2 b), the overall architecture has three components: 1) the spatiaotemporal embedding that summarizes object-level signals into a hidden representation h_t^k, 2) the amodal mask completor that takes h_t^k and V_t^k to output the estimated amodal mask $\tilde{M}_t^k$, and 3) the estimated amodal mask motion $\Delta\tilde{M}_t^k$. The generator function `Warp` itself does not contain any parameters.

Spatiotemporal embedding module This module extracts features from video frames, aligns and aggregates the feature across frames. The encoder (Enc) takes the concatenation of raw image patches, optical flow patches and visible masks as input. Then a recurrent architecture (Seq) aggregates information through temporal dimension.

$$f_t^k = \mathrm{Enc}\left(I_t^k, \Delta V_t^k, V_t^k\right) \tag{9}$$

$$h_t^k = \mathrm{Seq}\left(h_{t-1}^k, f_t^k\right) \tag{10}$$

h_t^k is the spatiotemporal embedding for object k at frame t. Here, we implement the encoder Enc and Seq with CNNs and LSTMs, respectively. This module also learns reasonable deformation over time.

Amodal mask completor Amodal mask completor is an Encoder-Decoder architecture with an information bottleneck. The CNN encoder takes the visible mask V_t^k and concatenate the output with h_t^k, then uses several de-convolutional layers DeConv to produce the full mask prediction:

$$\tilde{M}_t^k = \mathrm{DeConv}_a\left(\left[h_t^k, \mathrm{CNN}_a\left(V_t^k\right)\right]\right) \tag{11}$$

where a is the parameters of the CNN and DeConv above.

Motion predictor Computing $\Delta\tilde{M}_t^k$ uses the same general encoder-decoder architecture as the amodal mask completor except it takes ΔV_t^k instead of V_t^k as input. In addition, the computation takes the form of residual prediction, using the mean of visible mask motion signal $\Delta\bar{V}_t^k$ as the base to correct. This inductive bias reflect the common-fate principle of Gestalt Psychology[39].

$$\Delta\tilde{M}_t^k = \Delta\bar{V}_t^k + \mathrm{DeConv}_c\left(\left[h_t^k, CNN_c\left(\Delta V_t^k\right)\right]\right) \tag{12}$$

where c is the parameters of the CNN and DeConv above.

4 Experiments

We evaluate the proposed pipeline on both close-world setting with no distribution shifts between training set and test set, as well as the setting has distribution shifts with new object types.

Chewing Gum Dataset Chewing Gum is a synthetic dataset consists of random generated polygons (that look like chewing gum). Each polygon has a random number of nodes ranging from 7 to 12 and the nodes randomly scattered on a circle. Object are occluding each other and have relative movement. The occluded object never shows its full shape. But each part is shown in at least one frame. Because of the randomness in generation, the shape prior for a certain type will not work.

FISHBOWL Dataset This dataset [33] consists of 10,000 training and 1,000 validation and test videos recorded from a publicly available WebGL demo of an aquarium [1], each with 128 frames with resolution at 480×320. It is positioned between simplistic toy scenarios and real world data.

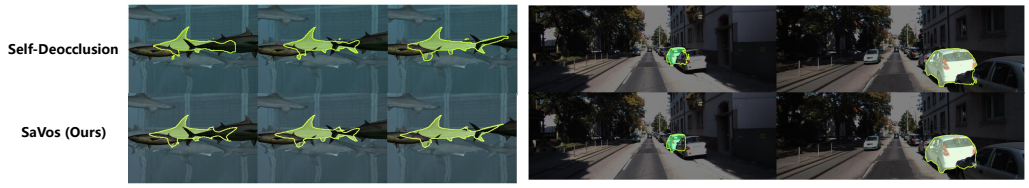

Figure 3: Qualitative comparison between the image-level baseline Self-Deocclusion [44] and our SaVos model. Our model produces more consistent predictions across frames and the performance is better when the occlusion rate is high. This is expected considering the consistency loss and the spatiotemporal architecture SaVos has.

KINS-Video-Car Dataset KINS is an image-level amodal dataset labeled from the city driving dataset KITTI [9]. In order to have SaVos work with KINS, we match images in KINS to its original video frame in KITTI. Since only training videos are available online, we re-split the original KITTI training set into three subsets for training, validation and test. We use PointTrack [42] to extract visible masks and object tracks to drive our video-based algorithm. We only run the algorithm for the `Car` category and mark this modified KINS dataset as `KINS-Video-Car`.

Metrics and Settings. The metric to evaluate amodal segmentation is mean-IoU as in [44, 33, 24]. Specifically, we compute mean-IoU against the groundtruth full mask as well as only the occluded sub-area, in order to evaluate the overall and focused performance. Occluded mean-IoU is usually a better indicator for amodal segmentation. We use groundtruth visible mask and tracking as inputs for FISHBOWL and Chewing Gum, and pre-compute visible mask and tracking from PointTrack[42] model for KINS-Video-Car. Note that self-supervision in this work is only for the amodal mask completion. On FISHBOWL, we only compute mean-IoU for objects with the occlusion rate from 10% to 70%. On KINS-Video-Car, we match the tracked objects and the annotated ones from KINS and only compute mean-IoU on the paired ones. All self-supervised models share the same input, while the supervised baseline is trained on samples with labels. For test-time adaptation, one test video is given and adapted separately. We run repeating experiments on our own method but not all baselines since the training is costly. The performance difference between runs is around 0.02 on the occluded mean-IoU. Paricularly, we further propose two new settings to evaluate the performance on distribution shifts. In the first setting, we train a model on four type of fishes in FISHBOWL and test on the rest type. In the second one, we train a model on FISHBOWL and evaluate on KINS-Video-Car.

Competitors. We use a simple heuristic method that just completes the object into a convex shape, a state-of-the-art image-level self-supervised model Self-Deocclusion [44], and supervised oracles: a recent state-of-the-art supervised method VRSP-Net [41] or U-Net [28], depending on the availability of the codebook for VRSP-Net.

4.1 Experiment Results on Test Set with No Distribution Shifts

Performance on FISHBOWL and KINS-Video-Car Table 1 compares SaVos with baselines. Qualitatively, Figure 3 compares predictions between SaVos and Self-Deocclusion. Note that KINS-Video-Car poses several challenges: the model-inferred object visible mask and tracking are inevitably inaccurate; videos with camera motion bring complex temporal motion signals like zoom-in/out, lens distortion and change of view point. However, our model still works on this challenging scenario. Also, some pixels for target car in the video of Figure 3 b) are never visible in any frames. This is a representative case for the parked cars that line up along roadside with corners always invisible. Our model still produce complete amodal mask, which can not be achieved if only spatiotemporal prior is learnt. This is an indicator that Savos also learns type prior during training.

Performance on Chewing Gum dataset Chewing Gum is `classless` as every two objects are different. As shown in Table 1, all image-level models fail to predict the occluded part. Since there is no type prior, no image-level model, including the supervised one, can do much better than completing the object to its convex hull. SaVos outperforms the rest with a significant advantage. Note that the occluded part only occupies a small portion of the entire full mask, thus mean-IoU on the occluded part is a better indicator on the model performance. Visualization is shown in Figure 4.

Table 1: Mean-IoU on FISHBOWL, KINS-Video-Car and Chewing Gum datasets. For the supervised oracle results, we use VRSP-Net[41] for KINS-Video-Car, and U-Net[28] for FISHBOWL and Chewing Gum since the codebook for VRSP-Net is not available for these two datasets.

Method	Supervised	FISHBOWL		KINS-Video-Car		Chewing Gum	
		Full	Occluded	Full	Occluded	Full	Occluded
Convex	✗	0.7761	0.4638	0.7862	0.0829	0.9264	0.3182
Self-Deocclusion [44]	✗	0.8704	0.6502	0.8158	0.1790	0.9624	0.3307
SaVos (Ours)	✗	**0.8863**	**0.7155**	**0.8258**	**0.3132**	**0.9746**	**0.8046**
Supervised Oracle	✓	0.9162	0.7500	0.8551	0.4883	0.9613	0.3321

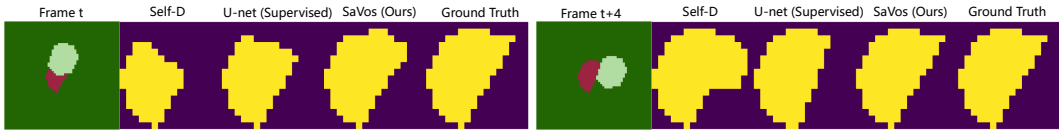

Figure 4: Amodal prediction on a Chewing Gum video on different frames. SaVos predicts consistent masks while for the image-based method the prediction varies even for the supervised method.

4.2 Experiment Results on Test Set with Distribution Shifts

It's a major challenge for machine learning models to generalize under distribution shifts and test videos may contain objects from unseen types. Amodal methods depending only on type prior suffers from these challenges, while a model that considers spatiotemporal prior can work better. When the model learns both type and spatiotemporal prior, test-time adaptation strategy can pick-up spatiotemporal prior to achieve good generalization performance. We verify this statement in the following experiments.

Test-time adaptation performance on FISHBOWL dataset with unseen fishes Trained on 4 types of fish, SaVos tries to fit a new type of fish out of the ones it has already learned (Figure 5). This is an indicator SaVos also learns type prior. Savos with test-time adaptation produces competitive result compared with a model trained with all types of fishes (Table 2). The image-level baseline model, although also test-time trained, doesn't show the same amount of improvement. This demonstrates that SaVos has picked up spatiotemporal prior as designed.

A more challenging scenario: adapting the model trained on FISHBOWL to KINS-Video-Car dataset. To further check the ability of adaptation on visually complex data, we use a model trained on FISHBOWL dataset and adapt it to the KINS-Video-Car dataset at test-time. Under this setting, the model is challenged to adapt to new object type, camera motion, new image quality as well as visual details. In Table 3, with test-time adaptation, our model even outperforms the image-level baseline trained directly on KINS-Video-Car, since our model leverages spatiotemporal prior, which leads to better generalization ability when distribution shift happens.

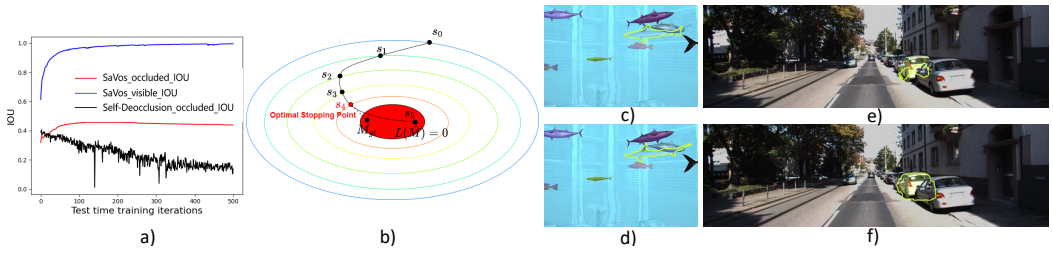

Figure 5: a) Learning curve for test-time adaptation on one FISHBOWL video.b) Learning dynamics for SaVos test-time adaptation. The optimal stopping point that is closest to the groundtruth might not be the one has the lowest loss. c) - d) SaVos first fits the unseen type of fish to a type seen in training time. After test-time adaptation, the incorrect type prior is gone. e) - f) Amodal performance before and after test-time adaptation for SaVos trained on FISHBOWL and test on KINS-Video-Car. In this case, though the model never see a car during training, it still make reasonable predictions. Though it captures less detail and slightly invades the neighboring pixels compared to a model trained on KINS-Video-Car in Figure 3 b).

Table 2: Methods with (w.) or without (w./o.) test-time adaptation (TTA). We report mean-IoU performance on unseen fishes. SaVos shows the advantage after applying test-time adaptation. S-D: Self-Deocclusion. Occ: occluded.

Unseen	SmallFishA	MediaFishA	MediaFishB	BigFishA	BigFishB	Overall	
Full/Occ	Occ	Occ	Occ	Occ	Occ	Full	Occ
Self-D	0.4757	0.6207	05825	0.5965	0.5709	0.8459	0.5859
Self-D w. TTA	0.4629	0.6349	0.6268	0.5742	0.5944	0.8497	0.5971
SaVos w./o. TTA	0.5362	0.6678	0.6044	0.5931	0.4978	0.8428	0.5905
SaVos w. TTA	**0.5464**	**0.7147**	**0.7080**	**0.7151**	**0.6230**	**0.8663**	**0.6886**
SaVos w. All	0.6299	0.7324	0.7256	0.7115	0.6894	0.8863	0.7155

Table 3: Adaptation performance from FISHBOWL training set to KINS-Video-Car test set. With test-time adaptation, SaVos trained on FISHBOWL achieves better mean-IoU compared with Self-Deocclusion trained on KINS-Video-Car.

Method	Training	Adaptation	Full	Occluded
Self-Deocclusion	FISHBOWL	✗	0.7813	0.0658
	FISHBOWL	KINS-Video-Car	0.7999	0.0969
	KINS-Video-Car	✗	0.8158	0.1790
SaVos(Ours)	FISHBOWL	✗	0.8004	0.0994
	FISHBOWL	KINS-Video-Car	0.8235	0.2976
	KINS-Video-Car	✗	0.8258	0.3132

4.3 Analysis on The Effectiveness of Test-time Adaptation for SaVos

Test-time adaptation optimization dynamic Figure 5 a) shows the learning curve of test-time adaptation on one FISHBOWL video. The model is trained on other four types of fishes and tuned to adapt to the unseen type BigFishB in that video. For our SaVos model, we noticed the occluded IoU performance increases first and then slightly drops, while the mean-IoU on the visible part keeps improving. A similar observation has been made in Deep Image Prior [36] where on the image denoising task, the network first learns the denoised version of the input image and then reconstructs the noisy pixels later. We assume similar optimization dynamic on our case. As illustrated in Figure 5 b), the training loss in Equation 7 is a surrogate one that only optimizes the visible mask in the other frames. The ground-truth solution M_{gt} belongs to a manifold of points that have loss: $L = 0$. However, there are other points can get the same loss while not really achieving perfect amodal segmentation. Usually those points refer to the predictions that not only cover the full mask, but also intrude into the neighboring area that is never revealed in any frame. Type prior can help resolving this issue by telling the common shape of that type. Then for seen types we don't worry about that. When type prior is not available in test-time adaptation, we try to tackle this using **early stopping**. We just stop when it finishes recovering all the visible part in every frame, leaving less chance to make additional intrusive predictions. In the experiment in this paper, we stop the optimization if the visible-part IoU improves less than 0.01 in the last 10 iterations. We also visualize the prediction before and after test-time adaptation in Figure 5 c)-f).

Why test-time adaptation is efficient on SaVos? Seems any self-supervised model is suitable in test-time adaptation since the same loss can be used in the training phase and test-time adaptation phase. However, we also try test-time adaptation on Self-Deocclusion while we don't observe the same amount of test performance improvement as SaVos in Table 2, 3.

The assumptions to explain this are *sampling efficiency* and *lack of motion*. Self-Deocclusion randomly selects another object as the occluder and overlay it on top of the visible mask of the target occludee to get supervision, again randomly. However, we would argue that such method is useful given a large training set but *particularly* not that efficient in test-time adaptation on a single video. The area they produces training signal is on the visible part at the current frame, which no need to be completed in this frame anyway. Only if the same part is visible in another frame and the occluder happens to be overlapped on the same position, that the training signal is contributing to the current test time video. Considering the image resolution and occluder type, the chance to sample that training signal can be low. While for SaVos, we produce the training signal that is exactly on the occluded areas in this video by building amodal motion across frames. The efficiency on test-time adaptation setting is an unique perspective to compare different loss design. Though both

are self-supervised, SaVos is more efficient than Self-Deocclusion after combining with test-time adaptation.

5 Conclusions

We propose SaVos, a self-supervised video amodal segmentation pipeline that simultaneously models the completed dense object motion and amodal mask. Beyond type prior, on which the existing image-level models rely, SaVos also leverages spatiotemporal priors for amodal segmentation. SaVos not only outperforms image-level baseline on several synthetic and real-world datasets, but also generalize better with test-time adaptation.

6 Limitations and Future Works

We summarize several limitations and future works of the existing SaVos model:

- Variational method can be introduced to handle uncertainty on the occluded area, especially for articulated non-rigid objects.
- Inductive bias from 3D modeling can be introduced to handle more complex ego and object motions.
- Currently, we need to run visible mask segmentation and tracking beforehand to start SaVos. It would be valuable to extend SaVos to an end-to-end pipeline, even all in self-supervised way. This leads to a future work of combining SaVos with video structured generative models like SCALOR[16]. However, we empirically tried to utilize the SCALOR code on KINS-Video-Car and found out object discovery on this dataset is still too challenging for existing methods. We'll also catch up with the progress of object discovery.

All these future works can be built on top of the idea of utilizing spatiotemporal information to mine amodal supervision signal and find evidence for mask completion from the existing SaVos.

7 Negative Social Impact

SaVos runs on object tracking result. Tracking on cars or pedestrians might have privacy concern.

References

[1] Webgl demo of an aquarium, 2022.

[2] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[4] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[5] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[6] John Duncan. Selective attention and the organization of visual information. *Journal of experimental psychology: General*, 113(4):501, 1984.

[7] Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019.

[8] David Forsyth and Jean Ponce. *Computer vision: A modern approach.* Prentice hall, 2011.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[10] Miguel Granados, James Tompkin, K Kim, Oliver Grau, Jan Kautz, and Christian Theobalt. How not to be seen—object removal from videos of crowded scenes. In *Computer Graphics Forum*, volume 31, pages 219–228. Wiley Online Library, 2012.

[11] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019.

[12] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[13] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of neurophysiology*, 28(2):229–289, 1965.

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[15] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 614–630, 2018.

[16] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. *arXiv preprint arXiv:1910.02384*, 2019.

[17] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Occlusion-aware video object inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14468–14478, 2021.

[18] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems*, 31, 2018.

[19] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. *Advances in Neural Information Processing Systems*, 33, 2020.

[20] Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler. Variational amodal object completion. *Advances in Neural Information Processing Systems*, 33:16246–16257, 2020.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[22] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021.

[23] Benjamin Peters and Nikolaus Kriegeskorte. Capturing the objects of vision with neural networks. *Nature human behaviour*, 5(9):1127–1144, 2021.

[24] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019.

[25] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[27] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] Sara Sabour, Andrea Tagliasacchi, Soroosh Yazdani, Geoffrey Hinton, and David J Fleet. Unsupervised part representation by flow capsules. In *International Conference on Machine Learning*, pages 9213–9223. PMLR, 2021.

[30] Yihong Sun, Adam Kortylewski, and Alan Yuille. Amodal segmentation through out-of-task and out-of-distribution generalization with a bayesian model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2022.

[31] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[32] Yu Sun, XiaolongWang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

[33] Matthias Tangemann, Steffen Schneider, Julius von Kügelgen, Francesco Locatello, Peter Gehler, Thomas Brox, Matthias Kümmerer, Matthias Bethge, and Bernhard Schölkopf. Unsupervised object learning via common fate. *arXiv preprint arXiv:2110.06562*, 2021.

[34] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6268, 2020.

[35] ZHUOWEN TU, XIANGRONG CHEN, Alan Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *IJCV*, 2005.

[36] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[37] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[38] X. Wang, A. Jabri, and A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[39] Max Wertheimer. *On perceived motion and figural organization*. MIT Press, 2012.

[40] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3):463–476, 2007.

[41] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao. Amodal segmentation based on visible region segmentation and shape prior. *arXiv preprint arXiv:2012.05598*, 2020.

[42] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[43] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2040–2050, 2019.

[44] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3792, 2020.

[45] Ziheng Zhang, Anpei Chen, Ling Xie, Jingyi Yu, and Shenghua Gao. Learning semantics-aware distance map with semantics layering network for amodal instance segmentation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2124–2132, 2019.

[46] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1464–1472, 2017.

[47] Daniel Zoran, Rishabh Kabra, Alexander Lerchner, and Danilo J Rezende. Parts: Unsupervised segmentation with slots, attention and independence maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10439–10447, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We put the code as part of the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We will put the code as part of the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material for Self-supervised Amodal Video Object Segmentation

Jian Yao[1]* **Yuxin Hong**[2]* **Chiyu Wang**[3]* **Tianjun Xiao**[4]† **Tong He**[4]
Francesco Locatello[4] **David Wipf**[4] **Yanwei Fu**[2]† **Zheng Zhang**[4]
[1] School of Management, Fudan University
[2] School of Data Science, Fudan University
[3] University of California, Berkeley
[4] Amazon Web Services
{jianyao20, yxhong20, yanweifu}@fudan.edu.cn, wcy_james@berkeley.edu
{tianjux, htong, locatelf, daviwipf, zhaz}@amazon.com

We organize the Supplementary Material as follows:

- We conduct ablation studies on the designs of SaVos, showing how they come naturally with our insight to achieve amodal.
- As the necessity of $\mathcal{L}_C$ is observed, we have another dive into the consistency loss to explain its motivation.
- We provide theoretical analysis on the SaVos loss design, showing more insights on how SaVos get supervision signal by exploring spatiotemporal information.
- For the cases can not be handled by spatiotemporal prior, we use empirical experiments and analysis to verify that type prior can handle those.
- We provide details about the method for reproducing the experiments. Code is attached in the supplementary.
- We have more visualizations from SaVos model. Videos are also attached.

A Ablation Studies on The Designs of SaVos

A.1 Ablation Study by Each Design Component

In this ablation study shown in Table 1, we verified the performance gain brought by each design component of SaVos on both FISHBOWL and KINS-Video-Car. The last row is the default setting for SaVos, according to the tables: removing any of the consistency loss, temporal embedding or bi-directional prediction will make the performance drop.

Note that our SaVos is not a combination of tricks each brings a little portion of gain. Those designs come naturally with our insight to achieve amodal completion using video. Since we use video, the consistency loss across frames is derived, and temporal embedding is also an intuitive design choice. Bi-directional prediction is not a must-have design. When we can run on offline setting, this is a why-not choice. Though SaVos already works well on online setting without Bi-directional prediction and the number beats the image-level baseline Self-Deocclusion[7]. We plot the mean value of the alpha channel for the forward pass and backward pass separately at each timestamp in Fig 1a). The plot meets our expectation of the alpha channel distribution: the alpha channel value for the forward pass is low at the beginning, increasing rapidly to get out of the "cold" region from frame 0-20, then slowing increasing from 20-100, finally increasing rapidly from 100-120 since that is the

*Work completed during internship at AWS Shanghai AI Labs.
†Correspondence authors are Tianjun Xiao and Yanwei Fu.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 1: Ablation study on SaVos components

Consistency Loss	Bi-directional prediction	Temporal Embedding	FISHBOWL		KINS-Video-Car	
			Full	Occluded	Full	Occluded
✓	✓	✗	0.8785	0.6894	0.8214	0.2792
✗	✓	✓	0.8606	0.6447	0.8244	0.2982
✓	✗	✓	0.8749	0.6849	0.8231	0.2962
✓	✓	✓	**0.8863**	**0.7155**	**0.8258**	**0.3132**

Table 2: Ablations on the architecture and losses for SaVos

Architecture	Loss	FISHBOWL		KINS-Video-Car	
		Full	Occluded	Full	Occluded
Self-Deocclusion [7]	Self-Deocclusion [7]	0.8704	0.6502	0.8158	0.1790
SaVos	Self-Deocclusion [7]	0.8742	0.6826	0.8021	0.2040
SaVos	SaVos	**0.8863**	**0.7155**	**0.8258**	**0.3132**

"cold" region for backward pass. Combining predictions from both direction can help on the "cold start" issue.

A.2 Disentanglement of The Contribution from Architecture and Self-supervised Loss

Compared with the ablation in Table 1 which focus more on the components of SaVoc itself, this part compares with the related work by disentangling the influence of loss and architectures. According to Table 2, when both training with the self-supervision method in Self-Deocclusion [7], SaVos architecture introduces around 0.03-0.04 occluded mIoU performance gain. After replacing the loss into SaVos loss, the model achieves another 0.04 performance gain on FISHBOWL and 0.11 gain on KINS-Video-Car. Both architecture and loss contribute to the performance gain, while the loss contributes more for scenes like KINS-Video-Car.

A.3 Ablation Study on SaVos Model Inputs

SaVos takes image patch I, visible mask V and the optical flow patch for the visible part ΔV as inputs. As shown in Table 3, removing either flow input or image patch harms the performance. We analyze the results as follows:

Removing optical flow input Removing flow input hurts the performance by a large gap up to more than 0.1 occluded IoU. In SaVos model design, we need to warp the amodal prediction to the subsequent frame to get supervision. The warping function takes amodal flow as input. As described in supplementary section E, we crop out the objects and rescale them to 64x128. This operation may lose the information of object motion. Then in the current setting, without flow, the network cannot infer object motion from the input thus the pipeline does not work properly. If we do not apply crop and scale, but put the full resolution mask and image as input, ideally the network can infer the flow from the sequence but directly providing the flow would avoid some unnecessary heavy-lifting and let the network focus on learning the completion signal.

Removing image patch input SaVos still outperforms the baseline in Self-Deocclusion[7]. Image patch provides important information about which part of the object is occluded now. The visible mask left the occluded part as background but image patch might contain the pixels of another object. Then the network would know where to complete. Image patch can provide some photomatic guidance for the flow completion, making the warping more accurate to collect more accurate supervision signals.

A.4 Learning-based Versus Rule-based Spatiotemporal Modeling

SaVos learns spatiotemporal prior for amodal segmentation. A proper baseline is manually aggregating optical flow across frames into a complete object shape. The detailed algorithm is introduced in Alg 1. We also take the union of the forward and backward . As shown in the below Table 4, SaVos outperforms this temporal baseline.

Table 3: Ablations on the SaVos model inputs

Model	FISHBOWL		KINS-Video-Car	
	Full	Occluded	Full	Occluded
Without flow	0.8508	0.6188	0.7922	0.1525
Without image	0.8673	0.6634	0.8025	0.2172
Full	**0.8863**	**0.7155**	**0.8258**	**0.3132**

Table 4: Learning-based SaVos versus rule-based spatiotemporal aggregation

Model	FISHBOWL Occluded IoU	KINS-Video-Car Occluded IoU
Temporal aggregation baseline k=5	0.3860	0.1487
Temporal aggregation baseline k=3	0.3937	0.1522
SaVos	**0.7155**	**0.3132**

The most important drawback to this naïve baseline is that if some parts of the objects are never visible in the whole video, this baseline of only using temporal information has no chance to complete the mask on those parts. This significantly limits the performance of this baseline in general. In contrast, SaVos proposes properly learning the type prior, which can still complete the whole mask.

To complete the whole object, this suggested naïve baseline may need to aggregate multiple frames. Consecutive warping across multiple frames might be inaccurate using the predicted flow. To verify this statement, we notice that k=3 actually works better than k=5, considering k=5 contains more information. In contrast, our SaVos model does not demand the constraint of consecutive warping, but only warping to one nearby frame. Our completion signals across multiple frames are aggregated by the consistency loss automatically using the backpropogation chain rule. This is the advantage of our learning based method that makes the model more robust to flow noise.

In the rule-based baseline, there won't be flow value for the occluded area for the occludee object. One has to employ a rule to fill the flow value for the occluded area; otherwise it is impossible to warp the whole to the next frame. To this end, this proposed baseline, we use the mean value of the visible part. However this might be inaccurate. In SaVos, we simultaneously predict the amodal mask and the flow value in the occluded area, which might be better than this baseline.

Algorithm 1 Rule-based spatiotemporal aggregating method

Input V, ΔV ▷ Visible masks and flows for an object
Initialize $AV = V_{n-k}$ ▷ Aggregated visible mask as AV
Initialize $\Delta AV = \Delta V_{n-k}$ ▷ Aggregated optical flow as ΔAV
for $t = n - k + 1$ to n **do**
  $AV = Warp(AV, \Delta AV) \cup V_t$ ▷ Warp the aggregated mask to take the union with V_t
  $IV = (1 - V_t) \cap AV$ ▷ Identify the area completed by AV but invisible in t
  $\Delta AV = \Delta V_t \cup (mean(\Delta V_t) * IV)$ ▷ Update ΔAV, for the occluded part, use the mean to fill
end for

B Another Dive into The Motivation of The Consistency Loss $\mathcal{L}_C$

One possible further enhancement on the supervision signal is to compare $\hat{M}_t^k$ to V_l^k in all frames with $l \geq t$, considering different occluded areas might be revealed in different frames. In order to make such comparison, in principle we can supervise the model training with the following complete loss:

$$\mathcal{L}_{complete} = \sum_{k=1}^{K} \sum_{t=1}^{T-1} \sum_{l=t}^{T} d\left(\hat{M}_t^k, V_l^k\right) \tag{1}$$

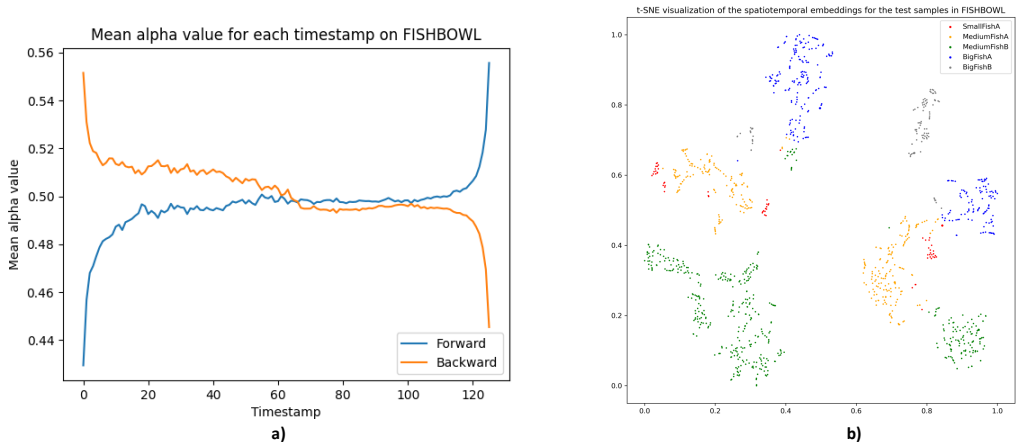

Figure 1: a) Distribution of the alpha channel of the bi-directional prediction. b) tSNE on the spatiotemporal embeddings for the test samples in FISHBOWL.

With some distance measure d. However, in practice this loss has too high computational cost, and the distance measure between $\hat{M}_t^k$ and V_l^k may be inaccurate when l is far from t.

Instead of directly comparing $\hat{M}_t^k$ and all V_l^k, we build a chain to accumulate the training signals along the sequence using the chain rule. To be specific, based on the definition of $\mathcal{L}_M$ and $\hat{M}_{t+2}^k$:

$$d_M(\hat{M}_{t+2}^k, V_{t+2}^k) = d_M\left(\text{Warp}\left(\tilde{M}_{t+1}^k, \Delta\tilde{M}_{t+1}^k\right), V_{t+2}^k\right) \tag{2}$$

and

$$\hat{M}_{t+1}^k = \text{Warp}\left(\tilde{M}_t^k, \Delta\tilde{M}_t^k\right) \tag{3}$$

After adding the consistency loss $\mathcal{L}_C$, we have penalty on $d_C(\tilde{M}_{t+1}^k, \hat{M}_{t+1}^k)$. Then the supervision signal from V_{t+2}^k will be passed through $\hat{M}_{t+2}^k \rightarrow \tilde{M}_{t+1}^k \rightarrow \hat{M}_{t+1}^k \rightarrow \tilde{M}_t^k$. So on so forth for $l > t+2$. This loss links the supervision signals in all future frames to guide the prediction in each frame.

C Theoretical Analysis on The Loss Design of SaVos

Theorem 1 *Given $\mathcal{L}_M^k = \mathcal{L}_C^k = 0$, with the following assumptions, we claim that the region covered by $\hat{M}_t^k$ includes M_t^k: 1) visible masks and object motion are from ground truth, 2) object k is rigid and simply connected, and 3) each pixel of this object is visible in at least one frame. Further, define an amodal trajectory $T_{\mathbf{p}}$ of point $\mathbf{p}$ as a sequence, $\{\mathbf{p}_t\}$ containing locations of a point $\mathbf{p}$ across frames, regardless of it being visible or occluded. We say $T_{\mathbf{p}} \parallel T_{\mathbf{q}}$ if $\mathbf{p}_t - \mathbf{q}_t$ is a constant for any t. Then, we claim that $\hat{M}_t^k = M_t^k$ with the following additional assumption: 4) Let $\mathbf{q}$ be an interior point of object k with its trajectory $T_{\mathbf{q}}$, any exterior point, p, of object k with its trajectory $T_{\mathbf{p}} \parallel T_{\mathbf{q}}$, appears in the background in at least one frame.*

This theorem makes an assumption that will not always strictly hold in practice, we emphasize that having global optima of the proposed loss is a necessary but not sufficient condition for aligning the correct amodal segmentation. The necessity indicates SaVos thoroughly captures the amodal supervision signals which could be explored from spatiotemporal information.

C.1 Proof of Theorem 1

We start from notations, and formal definitions of rigid object and visibility. Next, we discuss the behavior under the two losses and their insufficiency. Finally, we prove the theorem.

Notations Given that V_t^k represents the binary mask image for object k at frame t, we further use $\mathcal{V}_t^k$ to refer to the set of pixels being one in V_t^k. Similarly, $\mathcal{M}_t^k$, $\hat{\mathcal{M}}_t^k$, and $\tilde{\mathcal{M}}_t^k$, are sets of pixels being one in M_t^k, $\hat{M}_t^k$, $\tilde{M}_t^k$ respectively.

Assuming a constant resolution for all I_t, let $\mathcal{U}$ be the set of all pixels in any image I_t^k. Since for any pair of objects i and j we have $\mathcal{V}_{t+1}^i \cap \mathcal{V}_{t+1}^j = \emptyset$, we define the background region $\mathcal{B}_t$ as

$$\mathcal{B}_t = \mathcal{U} \setminus \bigcup_{i=1}^{K} \mathcal{V}_t^i = \bigcap_{i=1}^{K} \overline{\mathcal{V}_t^i} \tag{4}$$

with $\overline{\mathcal{V}_{t+1}^i}$ as the complement of $\mathcal{V}_{t+1}^i$. With the above additional notations, for object k with no occlusion at frame t we have $\mathcal{M}_t^k = \mathcal{V}_t^k$. For an object k being occluded at frame t we have $\mathcal{V}_t^k \subset \mathcal{M}_t^k$, which means the visible region is a subset of the amodal region, and $\mathcal{M}_t^k \setminus \mathcal{V}_t^k \subseteq \bigcup_{i \neq k} \mathcal{V}_t^i$, which means the occluded region belongs to the visible region of other objects.

Rigid object Let object k be a rigid object as in the assumption, its optical flow at any pixel is a pair of constant, specifically:

$$\begin{aligned} \Delta V_{\mathrm{x},t}^k[x,y] &= \Delta v_{\mathrm{x},t}^k \\ \Delta V_{\mathrm{y},t}^k[x,y] &= \Delta v_{\mathrm{y},t}^k \\ \Delta M_{\mathrm{x},t}^k[x,y] &= \Delta m_{\mathrm{x},t}^k \\ \Delta M_{\mathrm{y},t}^k[x,y] &= \Delta m_{\mathrm{y},t}^k \end{aligned} \tag{5}$$

for any $[x, y]$ in corresponding regions, where $\Delta v_{\mathrm{x},t}^k$, $\Delta v_{\mathrm{y},t}^k$, $\Delta m_{\mathrm{x},t}^k$, $\Delta m_{\mathrm{y},t}^k$ are four scalars for the optical flow, with $\Delta v_{\mathrm{x},t}^k = \Delta m_{\mathrm{x},t}^k$ and $\Delta v_{\mathrm{y},t}^k = \Delta m_{\mathrm{y},t}^k$.

As a result, its groundtruth amodal masks at frame t and $t+1$ satisfy $M_{t+1}^k[x + \Delta m_{\mathrm{x},t}^k, y + \Delta m_{\mathrm{y},t}^k] = M_t^k[x, y]$. Generally, for any pixel $[x_t, y_t] \in \mathcal{M}_t^k$, we can warp it to its counterpart $[x_l, y_l]$ in frame l since $[x_l, y_l] = [x + \sum_t^l \Delta m_{\mathrm{x},t}^k, y + \sum_t^l \Delta m_{\mathrm{y},t}^k]$.

Visibility The assumption on pixel visibility means that for any pixel $[x_t, y_t] \in \mathcal{M}_t^k$, there exists a frame l and a pixel $[x_l, y_l] \in \mathcal{V}_l^k$ such that $V_l^k[x_l, y_l] = M_t^k[x_t, y_t]$ and $[x_l, y_l] = [x_t + \sum_t^l \Delta m_{\mathrm{x},t}^k, y_t + \sum_t^l \Delta m_{\mathrm{y},t}^k]$.

Analysis on $\mathcal{L}_M$ We start with defining $\mathcal{W}_{t+1}^k$ as the set of pixels being one in W_{t+1}^k from the equation: $W_{t+1}^k = \left(\mathbf{1} - \sum_{i=1}^K V_{t+1}^i\right) + V_{t+1}^k$.

By definition, we have

$$\mathcal{W}_{t+1}^k = \mathcal{B}_{t+1} \cup \mathcal{V}_{t+1}^k \tag{6}$$

Thus $\mathcal{L}_M$ can be re-written as

$$\mathcal{L}_M = \sum_{k=1}^{K} \sum_{t=1}^{T} \mathcal{L}_M^{(k,t)} \tag{7}$$

where

$$\begin{aligned}\mathcal{L}_M^{(k,t)} &= \sum_{[x,y]\in\mathcal{W}_{t+1}^k} \text{BCE}(\hat{M}_{t+1}^k[x,y], V_{t+1}^k[x,y]) \\ &= -\sum_{[x,y]\in\mathcal{B}_{t+1}} \log\left(1-\hat{M}_{t+1}^k[x,y]\right) - \sum_{[x,y]\in\mathcal{V}_{t+1}^k} \log\left(\hat{M}_{t+1}^k[x,y]\right)\end{aligned} \tag{8}$$

For any k and t, $\mathcal{L}_M^{(k,t)}$ becomes zero only when:

$$\hat{M}_{t+1}^k[x,y] = \begin{cases} 1 & \text{if } [x,y]\in\mathcal{V}_{t+1}^k \\ 0 & \text{if } [x,y]\in\mathcal{B}_{t+1} \\ arbitrary & \text{if } [x,y]\in\bigcup_{i\neq k}\mathcal{V}_{t+1}^i \end{cases} \tag{9}$$

Note that for pixels in $\mathcal{V}_{t+1}^k$ and $\mathcal{B}_{t+1}$, the optimal value of $\hat{M}_{t+1}^k[x,y]$ equals to the true value in $M_{t+1}^k[x,y]$. However, for any pixel in $\bigcup_{i\neq k}\mathcal{V}_{t+1}^k$, $\hat{M}_{t+1}^k[x,y]$ can take arbitrary value without being penalized. Therefore $\hat{M}_{t+1}^k[x,y]$ is not guaranteed to exactly equal to $M_{t+1}^k[x,y]$, and this demonstrate the insufficiency of loss one.

Analysis on $\mathcal{L}_C$ The assumption of rigid object simplifies the relation between $\tilde{M}_t^k$ and $\hat{M}_{t+1}^k$. Specifically the Warp operation becomes

$$\hat{M}_{t+1}^k[x+\Delta\hat{m}_{\mathrm{x},t}^k, y+\Delta\hat{m}_{\mathrm{y},t}^k] = \tilde{M}_t^k[x,y] \tag{10}$$

Again, we re-write $\mathcal{L}_C$ as

$$\mathcal{L}_C = \sum_{k=1}^{K}\sum_{t=1}^{T-1}\mathcal{L}_C^{(k,t)} \tag{11}$$

where

$$\begin{aligned}\mathcal{L}_C^{(k,t)} &= \sum_{[x,y]\in\tilde{\mathcal{M}}_{t+1}^k\cup\hat{\mathcal{M}}_{t+1}^k} d\left(\tilde{M}_{t+1}^k[x,y] - \hat{M}_{t+1}^k[x,y]\right) \\ &= \sum_{[x,y]\in\tilde{\mathcal{M}}_{t+1}^k\cup\hat{\mathcal{M}}_{t+1}^k} d\left(\hat{M}_{t+2}^k[x-\Delta\hat{m}_{\mathrm{x},t}^k, y-\Delta\hat{m}_{\mathrm{y},t}^k] - \hat{M}_{t+1}^k[x,y]\right)\end{aligned} \tag{12}$$

This formulation indicates that a zero $\mathcal{L}_C^{(k,t)}=0$ means the masks in $\hat{M}_{t+1}^k$ and $\hat{M}_{t+2}^k$ have the same shape, but potentially at different locations. When $\mathcal{L}_C^{(k,t)}=0$ for any frame $t\leq T$, we have that all $\hat{M}_t^k$ have the same shape. Formally it means for any $[x_t,y_t]\in\hat{\mathcal{M}}_t^k$ and any other frame l, there exists a pixel $[x_l,y_l]\in\hat{\mathcal{M}}_l^k$ such that $\hat{M}_l^k[x_l,y_l] = \hat{M}_t^k[x+\sum_t^l\Delta m_{\mathrm{x},t}^k, y+\sum_t^l\Delta m_{\mathrm{y},t}^k]$. We name this property as the Transitivity.

However, this loss doesn't constraint on the relation of $\hat{\mathcal{M}}_{t+1}^k$ to $\mathcal{V}_{t+1}^k$ or $\mathcal{B}_{t+1}$, thus it is not sufficient to recover M_{t+1}^k by $\hat{M}_{t+1}^k$.

Proof of $\mathcal{M}_t^k\subseteq\hat{\mathcal{M}}_t^k$ Assume $\left\{\hat{M}_t^k\right\}_{t=1}^T$ is a set of amodal mask predictions for a rigid object k that satisfies $\sum_t\mathcal{L}_M^{(k,t)}=0$ and $\sum_t\mathcal{L}_C^{(k,t)}=0$, we now proof $\mathcal{M}_t^k\subseteq\hat{\mathcal{M}}_t^k$ for any t.

Since $\sum_t\mathcal{L}_M^{(k,t)}=0$, we have $\hat{M}_t^k[x,y]=M_t^k[x,y], \forall[x,y]\in\mathcal{V}_t^k\cup\mathcal{B}_t$. with $\sum_t\mathcal{L}_C^{(k,t)}=0$, we also have the accurate optical flow scalars $\Delta\hat{m}_{\mathrm{x},t}^k=\Delta m_{\mathrm{x},t}^k$ and $\Delta\hat{m}_{\mathrm{y},t}^k=\Delta m_{\mathrm{y},t}^k$, by estimating the flow from pixels $[x_t,y_t]\in\mathcal{V}_t^k$ and $[x_{t+1},y_{t+1}]\in\mathcal{V}_{t+1}^k$.

With the visibility assumption, for an occluded pixel $[x_t, y_t] \in \left(\bigcup_{i \neq k} \mathcal{V}_t^i\right) \cap \mathcal{M}_t^k$, we know there exists a frame l such that $[x_t, y_t]$ can be warped to a visible pixel in another frame l. Combining this and the transitivity property when $\sum_t \mathcal{L}_C^{(k,t)} = 0$, we have $\hat{M}_t^k[x_t, y_t] = V_l^k[x_l, y_l] = \hat{M}_l^k[x_l, y_l]$. Since $[x_l, y_l] \in \mathcal{V}_l^k$, we know $\hat{M}_l^k[x_l, y_l] = M_l^k[x_l, y_l] = 1$. Thus, $\hat{M}_t^k[x_t, y_t] = M_t^k[x_t, y_t] = 1 \forall [x_t, y_t] \in \mathcal{M}_t^k$, and that means $\mathcal{M}_t^k \subseteq \hat{\mathcal{M}}_t^k$.

Proof of $\mathcal{M}_t^k$ = $\hat{\mathcal{M}}_t^k$ We have proved that $\mathcal{M}_t^k \subseteq \hat{\mathcal{M}}_t^k$. Here we further analyze the factors of having a non-empty $\hat{\mathcal{M}}_t^k \setminus \mathcal{M}_t^k$.

A pixel $[x_t, y_t] \in \hat{\mathcal{M}}_t^k \setminus \mathcal{M}_t^k$ cannot be in $\mathcal{B}_t$ since $\mathcal{L}_M = 0$, thus it has to be in the region $\left(\bigcup_{i \neq k} \mathcal{V}_t^i\right) \cap \overline{\mathcal{M}}_t^k$, i.e. on another object. With the transitivity property, there exists $[x_l, y_l] \in \hat{\mathcal{M}}_l^k$ such that $[x_l, y_l] = [x_t + \sum_t^l \Delta m_{\mathrm{x},t}^k, y_t + \sum_t^l \Delta m_{\mathrm{y},t}^k]$ in each frame l. Since $[x_t, y_t] \in \overline{\mathcal{M}}_t^k$, we have $[x_l, y_l] \notin \mathcal{M}_l^k$ for any frame l. As a consequence, for any frame l, we have similar conclusions as $[x_l, y_l] \in \hat{\mathcal{M}}_l^k \setminus \mathcal{M}_l^k$, $[x_l, y_l] \notin \mathcal{B}_l$, and $[x_l, y_l] \in \left(\bigcup_{i \neq k} \mathcal{V}_l^i\right) \cap \overline{\mathcal{M}}_l^k$.

On the other hand, for any pixel $[x_t, y_t]$, if it is in $\mathcal{B}_t$, or it could warp to a pixel $[x_l, y_l] \in \mathcal{B}_t$, then we have $[x_t, y_t] \notin \hat{\mathcal{M}}_t^k \setminus \mathcal{M}_t^k$, and also $[x_l, y_l] \notin \hat{\mathcal{M}}_l^k \setminus \mathcal{M}_l^k$ for all corresponding $[x_l, y_l] = [x_t + \sum_t^l \Delta m_{\mathrm{x},t}^k, y_t + \sum_t^l \Delta m_{\mathrm{y},t}^k]$. Therefore, if every pixel $[x_t, y_t] \in \overline{\hat{\mathcal{M}}_t^k}$ or any of its corresponding $[x_l, y_l]$ appears in the background, then $\hat{\mathcal{M}}_t^k \setminus \mathcal{M}_t^k = \emptyset$.

Because we made the assumption that no pixel out of $\mathcal{M}_t^k$ has relatively static motion to object k across all frames that never appears in the background, so $\hat{\mathcal{M}}_t^k \setminus \mathcal{M}_t^k \neq \emptyset$ violates the assumption. Therefore, we conclude with $\mathcal{M}_t^k = \hat{\mathcal{M}}_t^k$.

D Empirical Analysis on How SaVos Handles Cases Missed by Spatiotemporal Prior

As mentioned in the above section, having global optima of the proposed loss is a necessary but not sufficient condition for aligning the correct amodal segmentation. To make good prediction on the cases where the assumptions in Theorem 1 are not hold, other prior and inductive bias from the architecture and data should chime in. To be more specific, we empirically found type prior learned through SaVos architecture handles cases missed by spatiotemporal prior. Typically, type prior models the shape prototype and variations for a certain type of object. When the object type is recognized, the model selects the prototype and certain variation to achieve amodal segmentation based on the context. The cases missed by spatiotemporal prior are usually the ones break the assumptions in Theorem 1. For example, part of an object keeps unseen in the whole video, as shown in Figure 2, where the car marked in blue behind the closest-in-path-vehicle (relative to ego) has the bottom left part always invisible. SaVos still complete the full amodal mask of the whole car.

Quantitatively, we split KINS-Video-Car into two parts. One presumably contains cases that break the "pixel-wise visible in video" assumption. Specifically, we collect 2899 out of 4644 cases of which the visible masks are touching with other objects in all tracked frames. This criterion roughly picks out the desired cases (roughly 70% of the selected cases break the assumption). In Table 5, the Mean-IOU on the occluded part for this subset is not significantly lower than the numbers for the whole set and the other subset. This validates SaVos has the capacity of handling more general amodal scenarios.

From the perspective of the architecture inductive bias, the encoder-decoder architecture in the Amodal Maks Completor contains an information bottleneck. This makes it easy to squeeze out type information since type is a concise representation and beneficial to amodal prediction. By receiving amodal supervision signals on different parts from different scenes, as long as the model learns to know they are amodal signals for the same type, those signals will be accumulated to form the type prior. This design choice also appears in several image-level methods such as [7] and [3]. In that case, SaVos can work as well as image-level methods on the cases break the "pixel-wise visible in video" assumption as long as the part invisible in this video can show up on different instance for the same type in other videos. If such a loosened assumption does not hold, then there will be no supervision

Figure 2: The car marked in blue behind the closest-in-path-vehicle (relative to ego) has the bottom left part always invisible. SaVos still complete the full amodal mask of the whole car. This is an indicator that SaVos not only just learn spatiotemporal prior, but also type prior.

Table 5: Occluded-Mean-IOU on subsets of KINS-Video-Car

	Full set	Subset 1 (break the assumption)	Subset 2
Mean-IOU on occluded part	0.3132	0.3104	0.3250

signal for that part in the whole dataset, image-level baseline models will also fail. Empirically, this can be verified in Table 1. SaVos can beat the Self-Deocclusion[7] baseline even without Temporal Embedding module. In test-time adaptation scenario where new types of objects appear, the methods rely only on type prior will fail while SaVos still have the chance to provide amodal completion using spatiotemporal prior.

To further verify this statement, We run tSNE on the spatiotemporal embeddings for the test samples FISHBOWL in Figure 1b) and noticed the data shows a clear clustering pattern consistent with the type information. Though objects for the same type might be split into more than one clusters, data points from different classes usually won't be entangled. This is an evidence that type information are learned.

E Additional Details about the Method

E.1 Detailed Architecture Hyperparameters

We use CNN to encode visual input for Spatiotemporal Embedding Module, Amodal Completer and Motion Predictor. The CNN has 6 layers. Each layer is followed by Group Normalization[5] and leaky ReLU nonlinearity. The default hyperparmeters for CNN encoder are given in the following Table 6.

To aggregate features on the temporal dimension, we use LSTM with 256 hidden dimensions in the Spatiotemporal Embedding Module.

Table 6: Default hyperparameters for CNN encoders

Parameter	Setting
internal resolution	64×128
channels in hidden layers	32, 64, 128, 256, 256, 512
strides in hidden layers	2, 2, 2, 2, (1,2), 1
kernels in hidden layers	4, 4, 4, 4, (3,4), 4
padding in hidden layers	1, 1, 1, 1, 1, 0

The decoders in Amodal Completor and Correspondence Predictor are built symmetrically by using the reversed list of transposed convolutions and layer parameters. Additionally for the last layers we remove the normalization and Leaky ReLU and add the sigmoid nonlinearity.

E.2 Implementation Details

We implement our model using PyTorch [4], which has BSD-style license. We use FlowNet2 [1] to extract optical flow. For all datasets, We train the model with the Adam [2] optimizer with the learning rate of 0.0001. We train 20 epochs on Chewing Gum and FISHBOWL dataset, 200 epochs for

Table 7: Frame-level mean IoU

Method	FISHBOWL		KINS-Video-Car	
	Full	Occluded	Full	Occluded
Self-Deocclusion [7]	0.8704	0.6502	0.8158	0.1790
SaVos (Ours)	**0.8863**	**0.7155**	**0.8258**	**0.3132**

Table 8: Object-level mean IoU

Method	FISHBOWL		KINS-Video-Car	
	Full	Occluded	Full	Occluded
Self-Deocclusion	0.8678	0.6418	0.7931	0.1952
SaVos (Ours)	**0.8843**	**0.7111**	**0.8063**	**0.3312**

KINS-Video-Car dataset and remain the model with the lowest self-supervised loss on the validation set. The batch size are 64, 24 and 8 respectively. We train on an AWS g4dn.12xlarge EC2 instance with four T4 GPUs. The training time is about one hour for Chewing Gum dataset and 24 hours for both FISHBOWL and KINS-Video-Car dataset.

E.3 mIoU Metric Computing

For each frame in a video, we calculate the mean IoU of the objects in that frame, and then average over all frames in a video dataset. We use this frame-level mean IoU metric in the paper. We can also calculate the mean IoU for all the objects without considering frames, which can be referred to as object-level mean IoU. We provide the results obtained by the above two metric computing methods in Table 7, 8 and comparing with the baseline, showing that the difference of these two metrics is small and the conclusions made in this paper are hold in either metric. Our released code provides both metrics.

F More Visualizations from SaVos

F.1 Visualization on KINS-Video-Car

The visualization on KINS-Video-Car in Figure 3 is mainly to show SaVos's performance on cars with different viewing angles and occlusion patterns. KINS-Video-Car is particularly challenging considering there is ego-camera model. Also, even use off-the-shelf SOTA segmentation and tracking model, the predicted visible mask and tracking still can't be perfect. SaVos still be able to complete object mask from different views of the cars.

Figure 3: Visualization of the prediction of our SaVos model on KINS-Video-Car dataset. The transparent mask is the predicted modal mask by PointTrack[6]. The solid curve are the contours of the amodal prediction by SaVos. From the figure we can see our model can predict the amodal mask well in the scene with multiple cars and heavy occlusion. Our model performs well for cars in different viewing angles and occlusion patterns. For examples: a)-d), which are cropped from the case 1)-4), respectively.

F.2 Visualization on Chewing Gum

Chewing Gum is a classless dataset. The image-level models fail since there is no type prior to be learned, even for the supervised method. SaVos learns to aggregate the temporal information from the video to achieve amodal mask completion in this dataset.

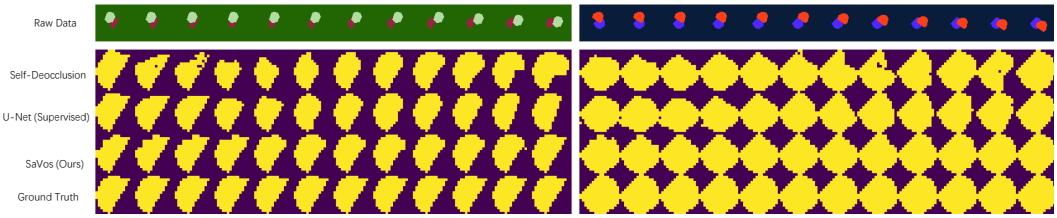

Figure 4: Visualization of the prediction on Chewing Gum dataset. We show the sequence for the images and the corresponding predictions for the occluded objects. From the visualization we can see that the image-level models fail in this toy dataset, since there is no type prior to be learned. However, the amodal mask can be predicted from temporal information. As showed in the figure, the SaVos model learn to aggregate the temporal information from the video and predict the amodal mask well in this dataset.

References

[1] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler. Variational amodal object completion. *Advances in Neural Information Processing Systems*, 33:16246–16257, 2020.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[5] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[6] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[7] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3792, 2020.