# Learning Dynamic Network Using a Reuse Gate Function in Semi-supervised Video Object Segmentation

Hyojin Park
Seoul National University
wolfrun@snu.ac.kr

Jayeon Yoo*
Seoul National University
jayeon.yoo@snu.ac.kr

Seohyeong Jeong*†
Seoul National University
AIRS Company, Hyundai Motor Group
seo_hyeong@snu.ac.kr

Ganesh Venkatesh
Facebook Inc.
gven@fb.com

Nojun Kwak
Seoul National University
nojunk@snu.ac.kr

## 1. Introduction

In this supplementary material, we provide detailed explanations and further experimental results that we were not able to include in the paper due to the limited space.

- Detailed process on the template matching method
- Detailed process on extending our method to other semi-VOS frameworks
- Result on reusing the previous mask
- Further experiments with YouTube-VOS dataset and ablation study on $P_{gate}$ with DAVIS 17
- Qualitative examples of our method

## 2. Details in the Template Matching

In this section, we describe a detailed process of the template matching as shown in Fig. 1 and Eq. (1). The current and the previous information are used together for this module and the module identifies movement between adjacent frames. First, the module compares a template and the input using matrix multiplication to produce a displacement feature map, $Z_t$. Second, information in $Z_t$ and the current feature map are blended together to generate the final dissimilarity feature, $D_t$.

Here, $X_t$ is a concatenated feature of $f8_t$ from the feature extractor and the previous score map, $S_{t-n}$. $q(X_t)$ is generated by passing $X_t$ into several convolutional layers as shown in Fig. 1, and it becomes $q(X_t) \in \mathbb{R}^{c_{tp} \times H \times W}$, where $c_{tp}$ is the number of channels of the feature map. In order to produce the dissimilarity feature map, we generate $Z_t$, which is the result of matrix multiplication between the template, $TP$, and a query feature map, $q(X_t)$, as follows:

$$Z_t = TP \times q(X_t). \tag{1}$$

$TP \in \mathbb{R}^{N_{tp} \times c_{tp}}$ is a fixed template matrix which consists of $N_{tp}$ embedding vectors of size $1 \times c_{tp}$ for representing the target object. The template is generated from the given initial frame and the corresponding ground truth mask by using the self-attention [5] in the initialization step.

$f8_t$ is forwarded to convolutional layers to produce modified feature map, $f8'_t \in \mathbb{R}^{c_f \times H \times W}$, where $c_f$ is the number of channels of the feature map. After then, $Z_t$ and $f8'_t$ are concatenated together to blend both information for creating the final displacement feature map, $D_t$. In this process, both $f8'_t$ and $D_t$ have the same resolution of $(H, W)$ and the same channel size of $c_f$. Finally, $D_t$ is used for both the gate function and the delta-generator.

## 3. Extension to Other Frameworks

In this section, we provide further explanation regarding Sec 3.3 of our main paper. To show that the proposed method can be used flexibly with other frameworks, we introduce how our method can be applied to not only FRTM but also to other semi-VOS frameworks. Fig 2(a) briefly describes the concept of our method. $F_t$ is a desired location for skipping layers in the feature extraction procedure. Likewise, $R_t$ is a desired location for skipping layers in the refined small feature map for making an accurate target mask. When the reuse gate is on, layers in sub-network after $F_t$ and before $R_t$ are skipped.

Fig 2(b) is a simplified framework on semi-VOS when the reuse gate is off. A process of semi-VOS is divided into 1) feature extraction, 2) localizing the target from the input by using target information, and 3) refining a small feature

*Indicate same equal contribution as second authors
†This work was done when Seohyeong Jeong was with SNU

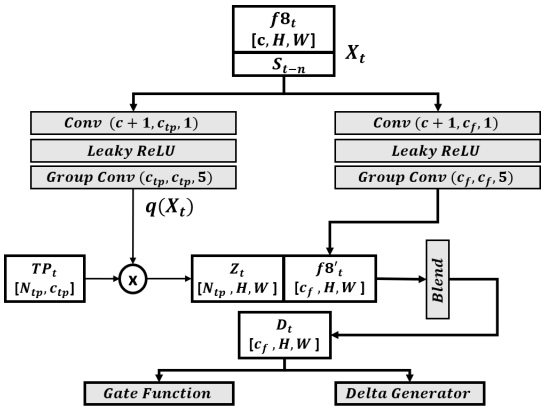


Figure 1. Process of template matching. The output feature map of the template matching module focuses on misalignment information between the current and previous frames. The output of template matching is forwarded to the gate function and the delta-generator.

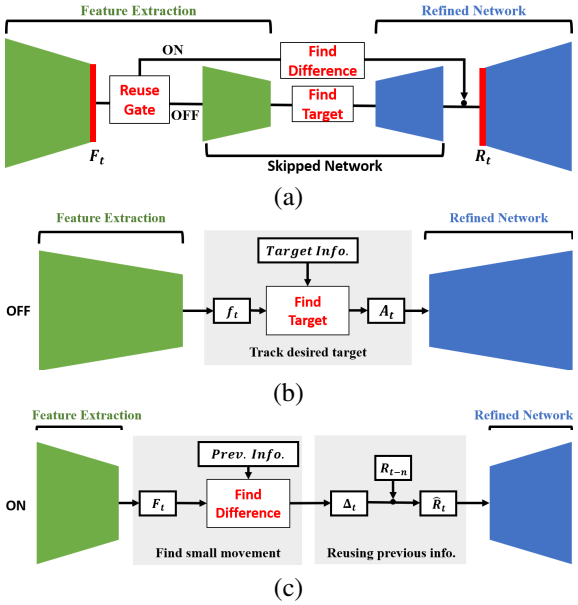


Figure 2. (a) Concept of architecture with applying the reuse gate function for semi-VOS frameworks. (b) Simplified semi-VOS framework. When the reuse gate is off, the model use original full-network. (c) Skipped semi-VOS framework. When the reuse gate is on, the model skips a sub-network and reuses previous features.

map for better mask generating quality. In detail, if feature extractor generates a small feature map, $f_t$, at frame $t$, the model finds a desired target from $f_t$ using target information which is generated in the initialization step. The target information can be embedded by a template features [5, 9, 8], memories [2, 4], fine-tuned weights [7, 3, 6] or parametric distribution [1] according to each model's approach. More specifically, FRTM [7] learns target-specific information by fine-tuning two layers of a network and TTVOS [5] generates a template containing target-specific information using matrix multiplication. After finding the desired target with the target-specific information, the model produces an attention map, $A_t$, which has an activation of the location of the target in pixel-wise level. Finally, the refined network elaborates the $A_t$ to match the resolution of the input frame and enhances fine-grained details.

Fig 2(c) explains how to skip sub-network from the original framework in Fig 2(b). Once the gate function decides to skip layers, the network reuses the feature map from the previous frame. To enable this, the model detects whether the frame has little or no movement little or not?? by comparing $F_t$ with the previous information. In our implementation, we apply the template matching method to find the difference and use score map to represent the previous information. For other models, it is possible to use the previous mask instead of the score map as previous information. After then, as we mentioned in our main paper, the model calculates difference between the current and the previous frames and generates $\Delta_t$. Finally, the model translates the previous refined feature map, $R_{t-n}$, using $\Delta_t$, to make $\hat{R}_t$. As mentioned above, to use the previous mask as previous information for training, $Loss_\Delta$ is calculated with $M_{t-n}$ instead of $S_{t-n}$ as follows:

$$Loss_\Delta = L2(\Delta_t, y'_t - M_{t-n}). \quad (2)$$

## 4. Skipping the Segmentation Network

The proposed method skips the segmentation network partially according to the output of the reuse gate function. We further devise a mask reusing method that skips the entire process of the segmentation network using the previous mask. Therefore, the overall architecture changes as shown in Fig 4. When the reuse gate is on, the model omits all layers remaining in the network. The refine-translator takes the previous mask, $M_{t-n}$, and $\Delta_t$ as an input and makes $M_t$ directly. The accuracy and FPS are described in Fig 3 along with proposed variation methods of **copy** and **fusion**, which are described in Fig. 7, Sec 4.2 of our main paper. **reuseM** is a method that reuses the previous mask. This method shows lower accuracy than others. We find that the model concentrates on increasing segmentation accuracy in training stage. Fig 5 shows the train accuracy and $Loss_{gp}$ during the training process in both originally proposed method of reusing previous features and the method of reusing previous masks. Compared to the original method of reusing features, the train accuracy shows comparable performance but the $Loss_{gp}$ does not converge to 0 in the method of reusing previous masks. This means that the model does not turn the reuse gate on properly as we intended in the training

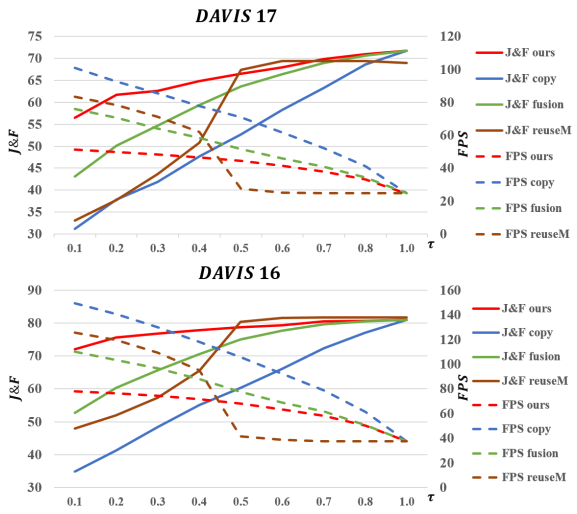


Figure 3. Further ablation study on different methods for reusing previous information when the reuse gate is on. Accuracy and FPS are reported on DAVIS datasets with different $\tau$ values. Ours is the proposed method based on FRTM-fast. **Copy** simply copies the previous mask for the current frame, without using the refine-translator. **Fusion** copies the previous mask if the similarity of consecutive frames is extremely large. Otherwise, original method of the refine-translator is used. **reuseM** is reusing the previous mask and it skips the score-generator and the segmentation network.

progresses. We conjecture that the reason of this outcome is due to not enough capacity in the refine-translator. The refine-translator takes the previous mask of the same size as an original input image instead of the intermediate feature map, $R8_{t-n}$. It incurs that the size of the receptive field is much smaller (by $1/8$) than in our original method or reusing features. Therefore, the model is learned to select the full path calculation due to the difficulty in reducing segmentation loss, when the reuse gate is on.

## 5. Further Experiment

In this section, we provide additional experiments on (1) Youtube-VOS with a different training scheme and (2) ablation study on $P_{gate}$ on DAVIS 17, which was not included in the main paper.

### 5.1. YouTube-VOS

We changed our training schemes from experiments in Sec 4.3. Firstly we changed the margin from $M(1, 0)$ to $M(0.5, 0)$. Secondly, we increase the number of epochs used in the training process by pre-training the model without proposed modules and by additionally training with our modules attached. Here, our modules include the reuse gate function, the template matching module, the delta-generator, and the refine-translator.

We provide following two reasons to the modification

| Method | reuseR | G | J | | F | |
|---|---|---|---|---|---|---|
| | | All | S | Us | S | Us |
| G-FRTM-fast ($\tau = 1$) | 0 | 63.8 | 68.3 | 55.2 | 70.6 | 61.0 |
| G-FRTM-fast ($\tau = 0.8$) | 25.5 | 63.4 | 67.6 | 55.8 | 69.3 | 60.9 |
| G-FRTM-fast ($\tau = 0.7$) | 40.0 | 62.7 | 67.1 | 55.2 | 68.2 | 60.1 |
| G-FRTM-fast ($\tau = 0.6$) | 50.9 | 62.3 | 66.7 | 55.3 | 67.2 | 60.0 |

Table 1. Quantitative comparison on YouTube-VOS benchmark validation set. **reuseR** denotes reusing rate. S and Us are seen and unseen categories. **G-** indicates using proposed method based on FRTM-fast.

we made. 1) Youtube-VOS Train set has less similar adjacent frames than DAVIS dataset as shown on Fig. 1(a) in Sec. 1 of our main paper. In our original setting, we use $m_1 = 1$, which means that as epoch continues, we force the model to reuse the mask more and eventually reuse at all time. However, here we give relaxation to the margin value (lower it to 0.5 from 1) to accommodate the characteristic of the dataset, Youtube-VOS. 2) When the gate is on, the model does not use a sub-network to make $R16_t$ and $R8_t$. Therefore, for the sub-network to get trained equally, we need training time for the model without the reusing process. The overall model accuracy with the proposed modification is better than without the pre-training stage (Tab. 3 in Sec. 4.3 of the main paper). Also, the accuracy of the unseen category has not changed with different values of the threshold.

### 5.2. $P_{gate}$ Experiment

Fig. 6 demonstrates that our gate function works properly following the ground truth similarity (IoU) between the current and the previous masks on multiple object scenario dataset, DAVIS 17. This experiment is equivalent to the experiment described in Sec 4.2 Fig. 9 but with DAVIS 17 instead of 16. $g_{iou}$ uses ground truth IoU instead of the estimated similarity probability from the gate function to decide whether to turn the reuse gate on or not. DAVIS 17 results on reuse rate and accuracy coincide with the experiments done with $g_{iou}$.

## 6. Qualitative Examples

We provide our qualitative examples on a single object (DAVIS 16) and multiple objects (DAVIS 17) settings. Fig. 7 shows an example of frame $23 - 26$ of the video *cows*. We finds that the proposed model shows better robustness than the original model. We conjecture that the template matching helps to discriminate the desired target objects from the non-target objects such as cow's legs from the fence. Fig. 8 shows that each object has different reuse rate depending on the movement. The reuse rate are 0.152 for the red segmented dog, 0.439 for the green segmented dog, and 0.864 for the yellow segmented human. Since dogs move faster and the human does not move as much as dogs.

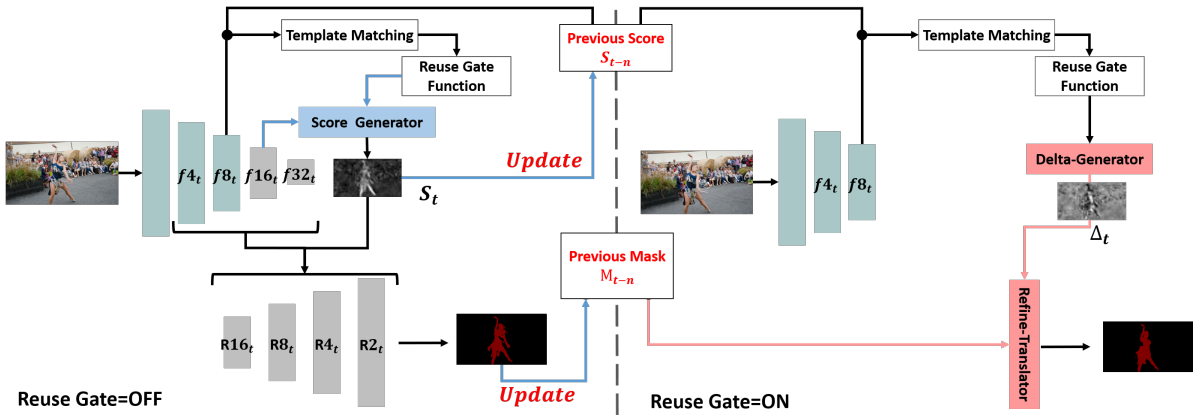


Figure 4. Overall architecture of another dynamic model. Some parts of feature extraction and all parts of segmentation network are skipped when the reuse gate is on. Refine-translator transforms the previous mask into the current one with the help of $\Delta_t$ to make the final mask.

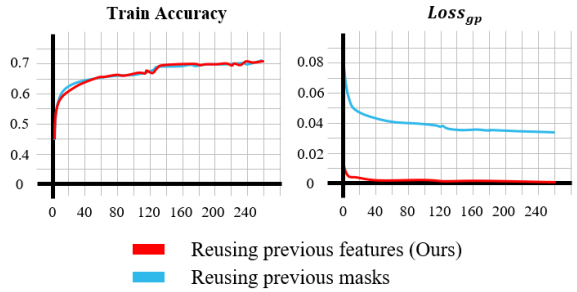


Figure 5. Graphs of the train accuracy and $loss_{gp}$ regarding reusing previous features (ours) *VS* reusing previous masks

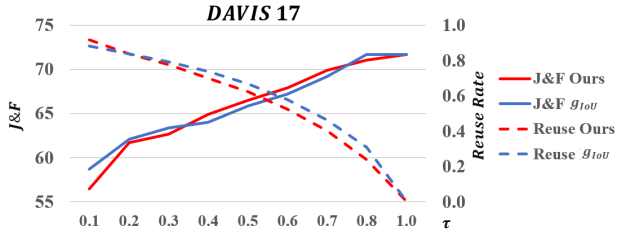


Figure 6. Ablation study on $P_{gate}$by comparison of accuracy and reuse rate on DAVIS 17 with different settings of $\tau$ . Ours estimates the similarity by gate function for deciding gate being on or off. $g_{IoU}$ is using ground truth IoU between adjacent frames as a similarity for deciding gate.

our gate function works properly depending on the movement of each object in the video.

## References


[1] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019.

[2] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *The European Conference on Computer Vision (ECCV)*, 2020.

[3] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1515–1530, 2018.

[4] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9226–9235, 2019.

[5] Hyojin Park, Ganesh Venkatesh, and Nojun Kwak. Ttvos: Lightweight video object segmentation with adaptive template attention module and temporal consistency loss, 2020.

[6] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2663–2672, 2017.

[7] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7406–7415, 2020.

[8] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019.

[9] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1338, 2019.

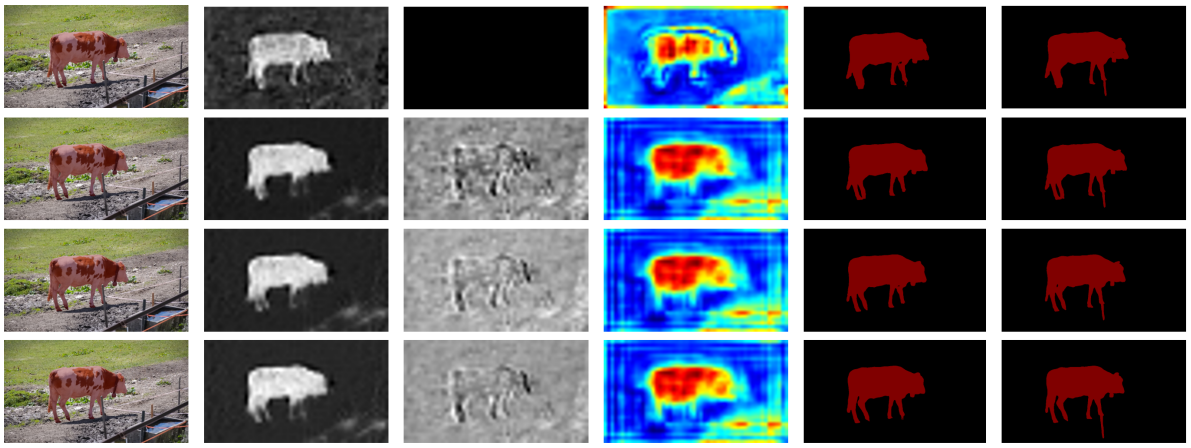


Figure 7. (a)-(f) Example of *cows* frame $23 - 26$. (a) Input frames are overlapped with ground truth masks. (b) $S$ and $\hat{S}$. Top of row is $S$ and the others are $\hat{S}$. (c) $\Delta_t$. The black image of top of row means this frame is not reused. Therefore, the $\Delta_t$ is not generated (d) $R8_t$ and $\hat{R8}_t$. Top of row is $R8_t$ and the others are $\hat{R8}_t$. (e) Our results (f) FRTM-fast results

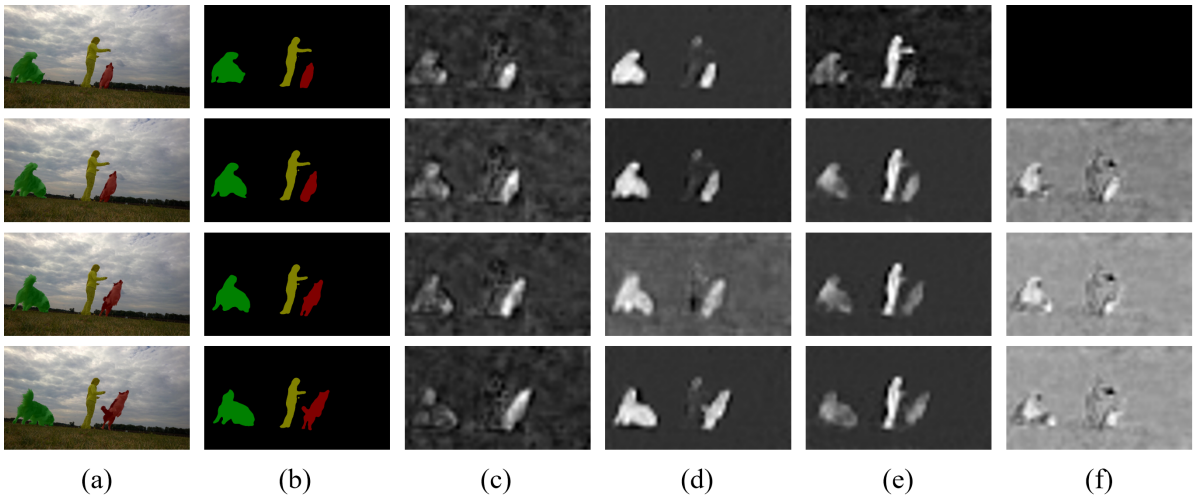


Figure 8. (a)-(f) Example of *dogs-jump* frame $29 - 32$. The red dog is object1, the green dog is object2 and human is object3. The reuse rates are 0.152, 0.439 and 0.864, respectively. (a) Input frames are overlapped with ground truth masks. (b) Our results (c) $S$ regarding to object1. All the frame does not reuse previous features (d) $S$ and $\hat{S}$ regarding to object2. Third of row is $S$ and the others are $\hat{S}$. (e) $S$ and $\hat{S}$ regarding to object3. Top of row is $S$ and the others are $\hat{S}$. (f) $\Delta_t$ regarding to object3. The black image of top of row means this frame is not reused. Therefore, the $\Delta_t$ is not generated