Video Patch Pruning: Efficient Video Instance Segmentation via Early Token Reduction

Supplementary Material

A. Training setting

This section provides a detailed specification of the training configurations employed in our Video Patch Pruning (*VPP*) framework.

For the Video Instance Segmentation (*VIS*) task we use ROVIS [43] as tracking method. It requires only two temporally corresponding frames during training, significantly lowering the memory consumption. In all our experiments we stick to the training settings, reported by ROVIS. Therefore we first train the dense model for 6 epochs using AdamW as optimizer with a weight decay of 0.05 and a learning rate of 2.5×10^{-5}, with a lr-decay of 0.1 after 4 epochs. For initialization, we use the model weights, pretrained on COCO dataset. In the second step, the patch pruning method is applied to the dense model using the standard training procedure. For consistency, this same protocol is maintained across all evaluated pruning methods to ensure a fair comparison.

VPP-specific hyperparameters are defined in Tab. 6. We utilize layer 6 from the previous frame for mapping. According to Fig. 2, x_6 shows almost maximal *Foreground Selectivity (FGS)* while remaining comparably dense, as shown in Fig. 5. Although deeper layers marginally improve *FGS*, they provide fewer features for reference, preventing highly sparse regions from having adequate background representatives for mapping. Moreover, we set the top-k selection of Map-SM module to 0.7 and 0.6 for Patch Keep Ratio (PKR) 55% and 40%, respectively. According to Dynamic-Vit [31] and TPS [37], we define the target keep ratio κ_l for each pruning stage as geometric sequence $[\rho, \rho^2, \rho^3]$. We set ρ to the value, where the goal PKR of 55% and 40% is reached.

name hyperparameter	value
pos idx Map-SM	1
ref-layer Map-SM	6
pos idx SM	3,6,9
τ	10
scale $\mathcal{L}_{sp}^{Map}$	10
scale $\mathcal{L}_{sp}^{SM}$	40
κ_{init}	0.5

Table 6. Hyperparameters used in Video Patch Pruning

In addition, we investigate the influence of the initial pruning threshold κ_{init} in Table 7. These experiments were conducted using ViT-Adapter Tiny on the YouTube-VIS 2021 dataset, optimized towards the goal-PKR of 55%. As defined in Eq. 6, κ_{init} specifies the intended mean patch probability $p^t \in \{0, 1\}$ to be reached by the *Map-SM* during training. Given that *Map-SM* utilizes a top-k selection mechanism, the training objective κ_{init} does not represent the sparsity level, but rather ensures that the estimated patch probabilities p^t remain within a stable, non-saturated range. The results show $\kappa_{init} = 0.5$ performs best in our setting, reaching an AP score of 42.3, which we therefore adopt as our default configuration.

κ_{init}	PKR (%)	AP↑
0.3	51.4	41.8
0.5	**52.7**	**42.3**
0.7	54.1	39.5

Table 7. Evaluation of model performance across different settings for κ_{init}. ViT-Adapter Tiny is trained on Youtube-VIS 2021 at a goal-PKR of 55%. $\kappa_{init} = 0.5$ shows the best performance.

B. Gumble Softmax function

In this section we define the Gumble-Softmax function[13], used in our proposed Map-SM module. For a set of class probabilities $\pi_1, \ldots, \pi_k$ the soft Gumbel-probability is defined as

$$\sigma_{gumbel}^{soft}(\pi_i, \tau) = \frac{exp(\frac{log(\pi_i) \cdot \tau + g_i}{\tau})}{\sum_{j=1}^{k} exp(\frac{log(\pi_j) \cdot \tau + g_j}{\tau})}, \tag{10}$$

where g_i denotes Gumbel-distributed noise. In this formulation temperature τ scales the Gumbel-distributed noise g_i, such that higher values minimize its stochastic influence on the resulting soft probabilities. Consequently, sampling the remaining patches with a higher τ, results in a reduced activation of background patches, limiting the reference features required for consistent background mapping. The addition of Gumbel-distributed noise g_i serves as a reparameterization trick, establishing a differentiable approximation of the discrete $argmax$ operation. This continuous relaxation enables the model to optimize categorical selections through standard backpropagation during the training phase. To bridge the gap between continuous probabilities, represented by σ_{gumbel}^{soft}, and discrete selections, the final mask is derived through a one-hot operation as straight-through estimator:

$$\sigma_{gumbel}^{hard}(\pi_i, \tau) = \text{one\_hot}(\sigma_{gumbel}^{soft}(\pi_i, \tau)). \tag{11}$$

In contrast to standard inference procedures [20, 24] that employ a deterministic $argmax$ to eliminate stochasticity, we retain Gumbel noise during inference for background activation. This noisy activation is essential for both identifying new objects in the video sequence and ensuring the background being sparsely activated for mapping.

C. Foreground Selectivity

In this section, we provide further details on the experiments to Foreground Selectivity (FGS) from Sec. 3. This experiment evaluates the capacity of dense, intermediate features to accurately identify foreground patches. Such a property is essential for patch pruning, as during pruning the focus should be shifted towards the foreground patches that belong to object instances. Background patches contain less relevant information and can even negatively effect the prediction [34]. Specifically, in instance segmentation tasks, background patches do not belong to target classes and should not be segmented. Consequently, the ability to bypass these non-informative regions allows the model to reduce overhead.

We evaluate patch relevance by adding binary classification heads to each intermediate layer of the frozen model. This head is optimized to differentiate between patches that belong to object related foreground or background. The ground truth label is extracted from the ground-truth segmentation mask. To mitigate the effects of class imbalance, such as the disproportionate background-to-foreground ratio observed in the YouTube-VIS datasets, we utilize weighted CE loss [26] for optimization. Specifically, we monitor the ratio of foreground patches through a running mean, denoted as $\overline{r}_{fg}$. To balance the loss, we weight the background and foreground class for the classification loss by $w_{bg} = \overline{r}_{fg}$ and $w_{fg} = 1 - \overline{r}_{fg}$, respectively. During training we freeze the whole model, except the intermediate classification heads. The classifier are trained for 2 epochs, following the settings form Sec. A. The learning rate is decayed by a factor of 0.1 after the initial epoch.

For the patch pruning task, specifically in instance segmentation tasks, the primary objective is to ensure that all foreground patches are retained during propagation. Therefore we define Foreground Selectivity (FGS) as the binary classification accuracy between foreground and background patches within feature x_i, given layer index i.

The results in Fig. 2 of the main paper show that the features almost linearly improve in depth in terms of Foreground Selectivity. Thereby, 0.5 indicates the lower bound classification, of a random patch sampling. At index 0, an FGS score of 0.6 indicates that while low-level features possess some discriminative information, it is insufficient for robust foreground selection. The FGS-score almost linearly increases in depth until layer 9. Overall, these results indicate that robust foreground patch selection is only achievable in the deepest layers. Common patch pruning methods such as DynamicViT [24], EViT [18], TPS [36] and DPS [32] typically reduce features linearly at predefined intervals of layers 3, 6 and 9. This aligns with our experimental results, suggesting that reliable foreground-background separation is a property of mature, deep-layer features, making early-stage pruning at layers 3 and 6 potentially suboptimal.

PKR (%)	dataset Youtube-VIS	method	IoI	$IoI_{\underline{S}}$	$IoI_{\underline{M}}$	$IoI_{\underline{L}}$
55%	2021	SViT	73.7%	77.2%	68.8%	65.8%
		VPP (ours)	**82.3%**	**87.3%**	**76.2%**	**72.2%**
	2019	SViT	63.1%	67.9%	61.0%	58.5%
		VPP (ours)	**77.3%**	**84.4%**	**75.0%**	**71.4%**
40%	2021	SViT	62.7%	67.5%	56.0%	52.1%
		VPP (ours)	**73.4%**	**79.5%**	**65.9%**	**60.7%**
	2019	SViT	54.9%	60.6%	52.4%	49.4%
		VPP (ours)	**66.7%**	**75.8%**	**63.7%**	**58.8%**

Table 8. Intersection over Instance (IoI) over datasets Youtube-VIS 2019 and 2021. VPP shows superior patch coverage every setting, improving the IoI scores by $> 10\%$ compared to SViT.

D. Intersection over Instance

In the experiments to *Intersection over Instance* (*IoI*) in the main paper, we analyzed how many foreground patches are actually activated during inference.

In instance segmentation tasks, background patches must not be segmented and therefore are less relevant for te segmentation task. We aim to maximize the activation of patches belonging to an instance, while reducing the computed patches to a certain point. Therefore we see *IoI* as valuable metric, especially for instance segmentation tasks.

Tab. 8 shows the IoI-scores for for Youtube-VIS 2019 and 2021, given a sparsity level of 55% and 40% PKR. In addition, we report IoI-scores across three sized categories based on spatial coverage: small (S), representing instances with a spatial size $\leq 10\%$; medium (M), ranging from 10% to 20%; and large (L), for instances exceeding 20%. The results show, *VPP* has a higher IoI-score on all settings, compared to SVIT. Notably, *VPP* allocates a higher patch density to smaller objects. This strategy is reasonable due to the fact that larger instances typically exhibit higher spatial redundancy [5], allowing for accurate predictions with a relatively lower patch sampling rate.

To qualitatively demonstrate the increased focus on object-related patches in *VPP*, Fig. 8 visualizes the patch activation patterns of both *SVIT* and *VPP*. The image-based pruning method *SVIT* processes the first three layers as fully dense, only introducing sparsity in deepest layers. Consequently, irrelevant background patches are processed unnecessarily at least five times. In contrast, *VPP* identifies and removes background patches in earlier layers, effectively shifting the computational focus toward foreground

Figure 7. Examples for the initial pruning mask M_1^t, generated by Mapping Selective Module (Map-SM). This mask removes 40% of patches after the first transformer block. All masks demonstrate that patches from foreground objects are sampled more densely, while background regions are effectively pruned during the initial mask generation.

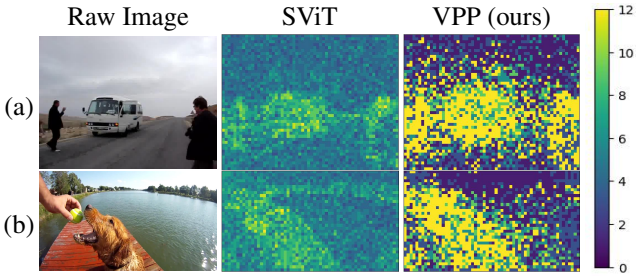

Figure 8. Comparison of token usage (55% PKR) for image-based (*SViT*) and video-based (*VPP*) Pruning. Unlike image-based patch pruning, *VPP* computes foreground features almost fully dense, while sparsely activating background patches for context preservation. For Video Instance Segmentation, this is essential to ensure the model focuses on the instances to be segmented, thereby minimizing redundant background computation.

regions. As illustrated in Fig. 8, foreground objects such as the person, dog, and hand exhibit nearly dense activation patterns.

E. Qualitative Pruning Results

In this section we show several examples for the applied pruning mask of the proposed *VPP* method. Therefore Fig. 7 demonstrates several examples for the initial pruning mask, generated by Map-SM and applied to the early ViT layers. Black patches are removed in the in layer 1 and are kept deactivated in all subsequent layers to save computational costs. The pruning masks consistently maintain high patch density on the foreground while preferentially removing background features. Nevertheless, *VPP* retains a sparse level of background activation, which is essential to identify new object instances. Note that *VPP* does not require auxiliary loss functions to enforce the retention of foreground patches during the pruning process. The optimization of instance segmentation losses [43] naturally biases the model toward foreground features, effectively prioritizing them over the background.

Moreover we show the patch activity of *VPP* in 8 exemplarly video sequences in Fig. 9 and 10. *VPP* highly activates foreground patches while removing background patches in early network layers. Note that all fully black patches are removed after layer 1 by the introduced *Map-SM*. The results show that moving objects, such as the cyclist in video 2, exhibit high spatial overlap with the pruning masks across all four stages, including the initial mask. This demonstrates that *Map-SM* effectively accounts for the temporal displacement of foreground patches throughout the video sequence.

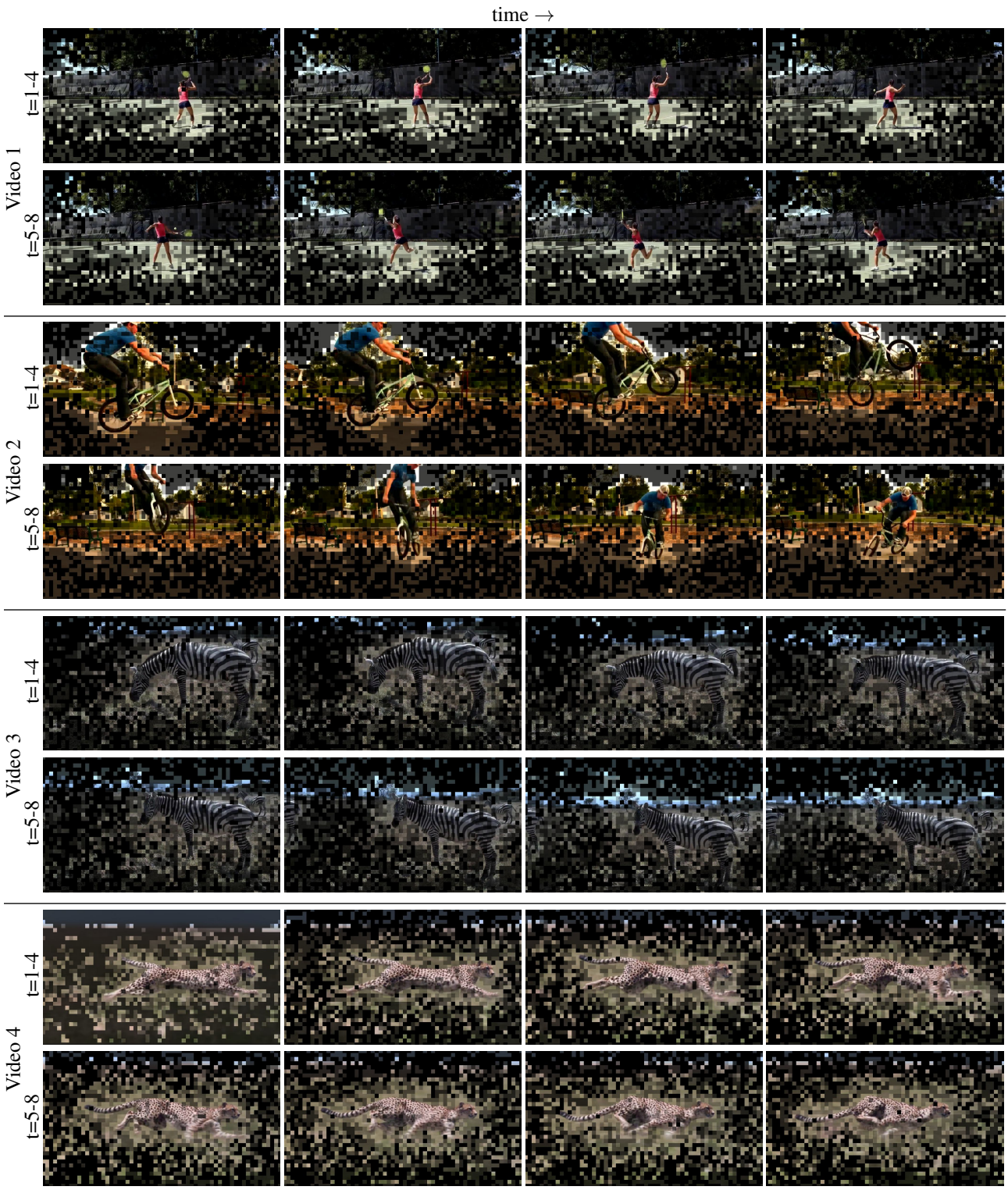

Figure 9. Exemplary pruning masks for videos 1–4 at a 40% Patch Keep Ratio (PKR). Results are shown for timesteps $t \in [1, 8]$. Patch activity is represented via grayscale shading, where darker patches are activated less. Fully black patches are removed by *Map-SM* after the first layer.

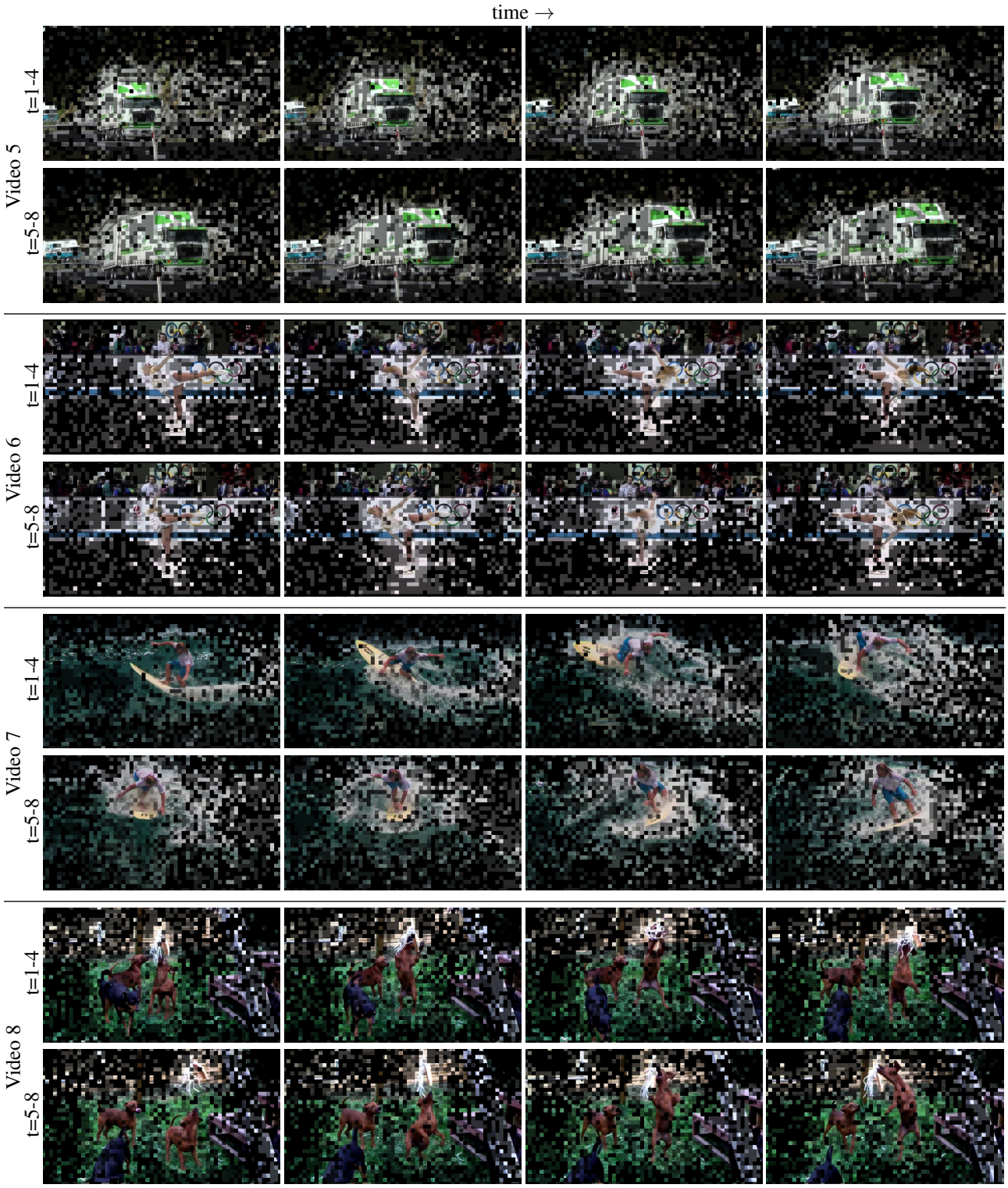

Figure 10. Exemplary pruning masks for videos 5-8 at a 40% Patch Keep Ratio (PKR). Results are shown for timesteps $t \in [1, 8]$. Patch activity is represented via grayscale shading, where darker patches are activated less. Fully black patches are removed by *Map-SM* after the first layer.