No More Sliding Window: Efficient 3D Medical Image Segmentation with Differentiable Top-K Patch Sampling

Young Seok Jeon[1][0000−0002−4948−083X], Hongfei Yang[2][0000−0002−8150−9364], Huazhu Fu[3][0000−0002−9702−5524], Yeshe Kway[4][0000−0001−6726−2425], and Mengling Feng[2⋆][0000−0002−5338−6248]

[1] Institute of Data Science, National University of Singapore, Singapore
youngseokjeon74@gmail.com
[2] Saw Swee Hock School of Public Health, National University of Singapore
{hfyang,ephfm}@nus.edu.sg
[3] Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
hzfu@ieee.org
[4] Oxford Centre for Clinical Magnetic Resonance Research, University of Oxford, United Kingdom
yeshekway@outlook.com

Abstract. 3D models surpass 2D models in CT/MRI segmentation by effectively capturing inter-slice relationships. However, the added depth dimension substantially increases memory consumption. While patch-based training alleviates memory constraints, it significantly slows down the inference speed due to the sliding window (SW) approach. We propose No-More-Sliding-Window (NMSW), a novel end-to-end trainable framework that enhances the efficiency of generic 3D segmentation backbone during an inference step by eliminating the need for SW. NMSW employs a differentiable Top-k module to selectively sample only the most relevant patches, thereby minimizing redundant computations. When patch-level predictions are insufficient, the framework intelligently leverages coarse global predictions to refine results. Evaluated across 3 tasks using 3 segmentation backbones, NMSW achieves competitive accuracy compared to SW inference while significantly reducing computational complexity by 91% (88.0 to 8.00 TMACs). Moreover, it delivers a 9.1× faster inference on the H100 GPU (99.0 to 8.3 sec) and a 11.1× faster inference on the Xeon Gold CPU (2110 to 189 sec). NMSW is model-agnostic, further boosting efficiency when integrated with any existing efficient segmentation backbones. The code is available: https://github.com/Youngseok0001/open_nmsw.

Keywords: Segmentation, Differentiable Top-K Sampling · Efficient Inference

⋆ Corresponding author: ephfm@nus.edu.sg

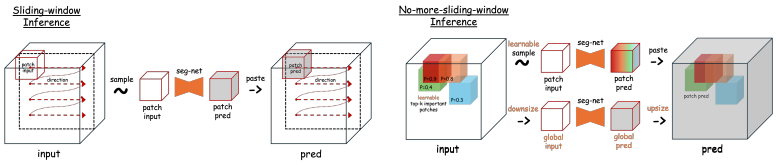

Fig. 1: NMSW provides an efficient alternative to computationally intensive SW inference. NMSW uses a differentiable patch sampling technique to select a handful of patches that enhance segmentation accuracy. NMSW incorporates coarse global prediction for the objects that require broader feature context.

1 Introduction

Patch-based models are slow and resource-heavy during inference: Scaling 2D models to 3D often leads to better segmentation performance in most 3D CT/MRI segmentation tasks [1, 10, 13]. However, training a 3D model with the full-res whole-body scan as input produces large intermediate tensors, which exceeds the GPU memory capacity.

Patch-based training, coupled with Sliding-Window (SW) inference, is the predominant method to address the substantial memory requirements of 3D segmentation models. Instead of processing the entire volume in a single step, patch-based training randomly samples patches that are significantly smaller than the actual volume. To generate the final whole-volume prediction using the patch-trained model, SW, as shown in Figure 1, is employed. SW makes sequential predictions on patches sampled at uniform intervals with some overlap (typically between 25-50% [6]). The overlapping predictions are then aggregated into the final whole-volume prediction with an appropriate post-processing step. Although SW is memory-efficient, computational efficiency and speed are greatly sacrificed. A typical ensemble UNet model [6] takes nearly a minute to perform a full SW on a whole-volume size of $512 \times 512 \times 458$ using an RTX 3090 GPU [23].

No-More-Sliding-Window: We introduce, for the first time in the community, a computationally efficient full-res CT/MRI segmentation framework, called the No-More-Sliding-Window (NMSW) which replaces the costly SW inference with a differentiable patch sampling module that learns to sample only a handful of patches of higher importance. NMSW aggregates the predictions from the selected patches with a low-res global prediction to produce the final full-res whole-volume prediction.

Specifically, NMSW operates through a three-step process: (1) **Global Prediction**: A global model processes a low-resolution whole-slide volume, generating two outputs: a coarse global segmentation prediction and a probability mass function (pmf) that indicates the likelihood of regions enhancing the final prediction score. (2) **Patch Selection and Prediction**: High-resolution patches are selected based on the region highlights sampled from the pmf using

our proposed Differentiable-Top-K sampling module. These selected patches are then processed by a local model to generate granular local predictions. (3) **Final Aggregation**: The coarse global prediction is combined with the Top-K patch predictions through our Aggregation module to produce the final prediction.

We emphasize that NMSW is not a new segmentation backbone but a framework designed to enhance the computational efficiency of existing 3D medical image segmentation backbones. Across evaluations on three multi-organ segmentation tasks using three different backbone models, NMSW consistently achieves competitive segmentation performance, and in some cases even surpasses the SW baseline, while reducing computational cost by 91%.

2 Related Works

Previous efforts to reduce the computational cost of 3D segmentation models can be broadly categorized into three approaches: (1) optimizing backbone architectures while retaining patch-based training and sliding window inference [9,17,18]; (2) reframing segmentation as a super-resolution task; and (3) allocating more computation to regions of interest—following a similar spirit as NMSW—though this approach struggles with generalizability and scalability.

In reducing the segmentation backbone complexity, [17] proposes to replace the self-attention block with a module that processes fewer input tokens. [18] apply knowledge distillation to train a compact student model under a larger teacher model. Although these methods streamline network architectures, they still rely on expensive SW inference. NMSW is compatible with these lightweight backbone, further improving inference speed computation cost.

[22] and [24] adopt super-resolution techniques to infer high-resolution segmentation maps from sparse, low-resolution inputs or features. Although these approaches alleviate the need for sliding window inference, they require heavy architectural modification and have not been validated on complex segmentation tasks. In contrast, NMSW has been evaluated on challenging multi-organ segmentation benchmarks and is expected to generalize well across other diverse tasks, as it leverages robust backbone architectures such as MedNeXt [21] and UNETR-Swin [5].

In an effort to allocate a dynamic computational budget, [4] downscales all objects to fit within a predefined patch size and subsequently upsamples them. However, this approach leads to information loss, resulting in suboptimal segmentation accuracy. [19] estimates a deformed grid to resample high-resolution inputs [7], dynamically adjusting the shape and size of objects in the input image. However, such deformation often introduces interpolation errors and leads to information loss. [15] employs Deep Q-Learning [16] to iteratively refine cropping regions based on ground truth overlap. While effective in some settings, this method is computationally inefficient and poorly scalable for multi-organ segmentation tasks due to the overhead of iterative 3D volume updates.

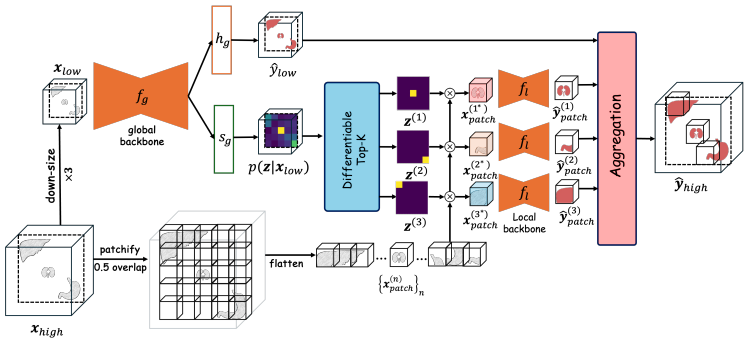

Fig. 2: NMSW takes the full-res scan $\mathbf{x}_{\text{high}}$ as an input. Global backbone produces coarse prediction $\hat{\mathbf{y}}_{\text{low}}$ and a patch importance pmf $\text{p}(\mathbf{z}|\hat{\mathbf{y}}_{\text{low}})$. Top-$K$ important patches $\{\mathbf{x}_{\text{patch}}^{(k^*)}\}$ are sampled based the pmf. $\hat{\mathbf{y}}_{\text{low}}$ and $\{\mathbf{x}_{\text{patch}}^{(k^*)}\}$ are aggregated to produce the full-res prediction $\hat{\mathbf{y}}_{\text{high}}$.

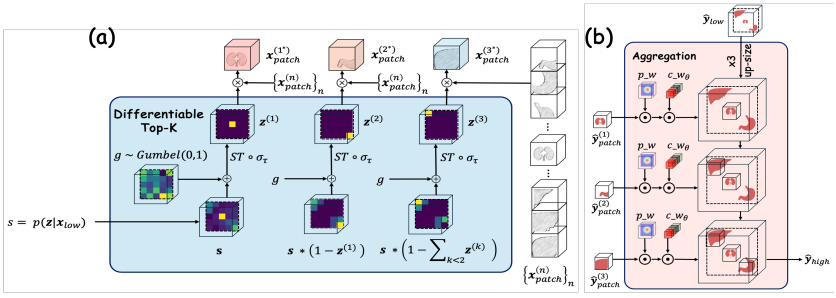

Fig. 3: Differentiable-Top-K block selects K important patches from a learned categorical distribution using the Gumbel-Softmax trick. The Aggregation block combines the global prediction and local patch predictions to generate the final prediction.

3 Method

As shown in Fig. 2, unlike conventional patch-based training, NMSW takes the entire scan, $\mathbf{x}_{\text{high}} \in \mathbb{R}^{H\times W\times D}$, as input. The input is mapped to a down-sampled scan $\mathbf{x}_{\text{low}} \in \mathbb{R}^{H'\times W'\times D'}$ and a set of overlapping patches, $\mathbf{X}_{\text{patch}} = [\mathbf{x}_{\text{patch}}^{(1)}, \mathbf{x}_{\text{patch}}^{(2)}, \ldots, \mathbf{x}_{\text{patch}}^{(N)}] \in \mathbb{R}^{N\times H_p\times W_p\times D_p}$, sampled at regular intervals where, N represents the total number of patches[5].

The global backbone f_g takes $\mathbf{x}_{\text{low}}$ and generates two outputs: (1) a coarse global prediction $\hat{\mathbf{y}}_{\text{low}} \in [0,1]^{C\times H_l\times W_l\times D_l}$ and (2) a discrete probability distribution $\text{p}(\mathbf{z}|\mathbf{x}_{\text{low}}) \in [0,1]^N$, which estimates the importance of individual patches in contributing to the final prediction $\hat{\mathbf{y}}_{\text{high}}$. Using the Differentiable-Top-K module, K important patch locations, $\{\mathbf{z}^{(k)} \in [0,1]^N\}_{k=1}^K$, are sampled with-

[5] The value of N is computed as $N = N_h \cdot N_w \cdot N_d$, where N_h, N_w, and N_d denote the number of patches along each dimension

out replacement from $\mathrm{p}(\mathbf{z}|\mathbf{x}_{\text{low}})$. These locations are used to select K important patches, $\{\mathbf{x}_{\text{patch}}^{(k^*)}\}_{k=1}^{K}$, from $\mathbf{X}_{\text{patch}}$.

The selected patches are then processed by a patch-level segmentation model f_l, which produces predictions $\{\hat{\mathbf{y}}_{\text{patch}}^{(k)}\}_{k=1}^{K}$. The global and local models do not share weights. Finally, the Aggregation block combines $\hat{\mathbf{y}}_{\text{low}}$ with $\{\hat{\mathbf{y}}_{\text{patch}}^{(k)}\}$ to generate the final whole-volume prediction, $\hat{\mathbf{y}}_{\text{high}}$.

Differentiable-Top-K: Training the Differentiable-Top-K module is challenging due to two non-differentiable operations: (1) stochastic sampling and (2) the discrete nature of the samples. To address this, we adapt the Reparameterizable Subset Sampling algorithm [11], which extends the Gumbel-Softmax trick to Top-K sampling scenarios.

We first introduce the Gumbel-Softmax trick [8, 14], which provides a continuous relaxation for sampling from a categorical distribution. Given a categorical distribution $z \sim \mathrm{p}(\mathbf{z})$, where the probability of the n-th outcome is $\mathrm{p}(\mathbf{z} = n) = \pi_n$, the Gumbel-Softmax approximates sampling as:

$$\mathbf{z}_{\text{softhot}} = [y_1, y_2, \ldots, y_N], \quad y_n = \sigma_\tau(\log(\pi_n) + g_n), \tag{1}$$

where $\operatorname{argmax}(\mathbf{z}_{\text{softhot}})$ yields z. Here, $g_n \sim \text{Gumbel}(0, 1)$ is a sample from the Gumbel distribution, and σ_τ is a softmax function with temperature parameter $\tau \in [0, \infty]$:

$$\sigma_\tau(x_n) = \frac{\exp(x_n/\tau)}{\sum_{m=1}^{N} \exp(x_m/\tau)}. \tag{2}$$

Differentiable-Top-K module (Fig. 3(a)) generalizes this approach to draw Top-K samples without replacement by masking the probabilities of previously sampled outcomes. The k-th sample, $\mathbf{z}_{\text{softhot}}^{(k)}$, is defined as:

$$\mathbf{z}_{\text{softhot}}^{(k)} = [y_1^{(k)}, y_2^{(k)}, \ldots, y_N^{(k)}], \quad y_i^{(k)} = \sigma_\tau(\log(\pi_i^{(k)}) + g_i), \tag{3}$$

where

$$\pi_i^{(k)} = \begin{cases} \pi_i^{(k-1)} & i \neq \operatorname{argmax}(\mathbf{z}_{\text{softhot}}^{(k-1)}), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Our application strictly requires the sampled variable to be onehot rather than softhot, as artifacts from other patches may otherwise contaminate the extracted patches. To turn $\mathbf{z}_{\text{softhot}}$ to a onehot sample $\mathbf{z}_{\text{onehot}}$, we employ a Straight-Through (ST) estimator[6]. However, ST introduces gradient bias during early training when probabilities are not saturated. To mitigate this, we propose to scale the onehot samples with their corresponding softhot values: $\mathbf{z}^{(k)} = \mathbf{z}_{\text{onehot}}^{(k)} \cdot \mathbf{z}_{\text{softhot}}^{(k)}$. This adjustment reduces gradient bias and accelerates convergence empirically. The Top-K patches are extracted via an inner product:

$$\mathbf{x}_{\text{patch}}^{(k^*)} = \langle \mathbf{z}^{(k)}, \mathbf{X}_{\text{patch}} \rangle, \quad \forall k \in \{1, \ldots, K\}. \tag{5}$$

[6] In PyTorch, ST is implemented as $\mathbf{z}_{\text{onehot}} := \mathbf{z}_{\text{softhot}}.\text{detach}() + \mathbf{z}_{\text{onehot}}.\text{detach}() - \mathbf{z}_{\text{softhot}}$.

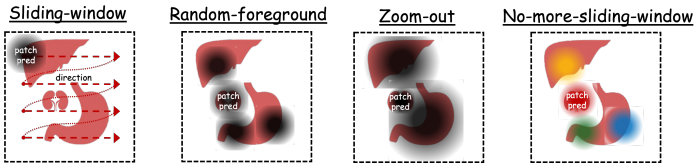

Fig. 4: Comparing Inference Methods.

Since $\mathbf{z}^{(k)}$ lies in the interval $[0, 1]$, the intensity of the extracted patches is proportionally scaled, as illustrated in Fig. 3(a) as the colored patches.

Our Differentiable Top-K module differs significantly from that of Cordonnier et al. [3] in both learning objective and module design. [3] simulates Top-K sampling using perturbed maximization [2], which yields soft-hot samples and results in blended patches. While such blending may be tolerable for 2D classification, it deteriorates the fine-grained features crucial for 3D segmentation. In contrast, our Gumbel-Softmax-based Top-K module produces clean, unblended patches better suited for segmentation task.

Aggregation: The Aggregation block (Fig 3(b)) merges patch predictions $\{\hat{\mathbf{y}}^{(k)}_{\text{patch}}\}$ with the upscaled global prediction $\hat{\mathbf{y}}_{\text{up}}$ to produce $\hat{\mathbf{y}}_{\text{high}}$. Each patch is weighted by $\mathbf{p\_w} \in [0, 1]^{P_h \times P_w \times P_d}$, a discretized Gaussian distribution with mean 0 and variance 0.125^2, to blend predictions in overlapping regions. Additionally, a learnable class weight $\mathbf{c\_w}_\theta \in [0, 1]^C$ is introduced to balance global and patch predictions. The final prediction in patch region $\Omega(k)$ is computed as:

$$\hat{\mathbf{y}}_{\text{high}}(\Omega(k)) := \sigma(\mathbf{c\_w}_\theta) \cdot \mathbf{p\_w} \cdot \hat{\mathbf{y}}^{(k)}_{\text{patch}} + (1 - \sigma(\mathbf{c\_w}_\theta)) \cdot \hat{\mathbf{y}}_{\text{up}}(\Omega(k)). \tag{6}$$

3.1 Loss Function

We optimize the model using the conventional soft-Dice loss combined with cross-entropy, applied to three key outputs: the low-resolution prediction $\hat{\mathbf{y}}_{\text{low}}$, patch-level predictions $\{\hat{\mathbf{y}}^{(k)}_{\text{patch}}\}_{k=1}^{K}$, and the final high-resolution output $\hat{\mathbf{y}}_{\text{high}}$. To promote diversity in patch selection during training, we incorporate an entropy regularization term $\mathcal{H}$ on the sampling distribution $\text{p}(\mathbf{z} \mid \mathbf{x}_{\text{low}})$. The complete objective function is:

$$\begin{aligned} \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Dice}}(\mathbf{y}_{\text{low}}, \hat{\mathbf{y}}_{\text{low}}) + \mathcal{L}_{\text{Dice}}(\mathbf{y}_{\text{high}}, \hat{\mathbf{y}}_{\text{high}}) \\ + \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\text{Dice}}(\mathbf{y}^{(k)}_{\text{patch}}, \hat{\mathbf{y}}^{(k)}_{\text{patch}}) + \lambda \mathcal{H}(\text{p}(\mathbf{z} \mid \mathbf{x}_{\text{low}})), \end{aligned} \tag{7}$$

where λ is a hyper-parameter to control the degree of exploration.

4 Experiments

Fig 4 illustrates the 3 baseline inference methods: SW, Random Foreground (RF) and Zoom-out. Similar to NMSW, RF samples patches containing objects

Table 1: Comparison of accuracy and efficiency between the NMSW and the baseline methods in three segmentation tasks. k represents the number of patches used for each inference. The speed of RF is omitted, as its network structure is nearly identical to NMSW. The Inference speed and MACs assumes an input of size 1 x 480 x 480 x 480 with 20 output channels. Model size of Zoom-out and RF is same as NMSW. **Best**, **2nd-best** and 3rd-best results are marked.

	Word		TotalOrgan		TotalVert		Speed			
Inference type	**DSC**	**NSD**	**DSC**	**NSD**	**DSC**	**NSD**	**GPU**	**CPU**	**MACs(T)**	**# Param(M)**
UNet										
SW (gold standard)	**0.852**	**0.906**	**0.868**	**0.904**	**0.884**	0.940	12.3	135	63.2	26.5
Zoom-out	0.837	0.888	0.856	0.890	0.869	0.933	5.19	144	**3.83**	-
RF (k=5)	0.790	0.806	0.789	0.802	0.684	0.765	-	-	-	-
RF (k=30)	0.829	0.870	0.832	0.856	0.773	0.851	-	-	-	-
NMSW (k=5)	0.825	0.867	0.841	0.870	0.832	0.908	**0.147**	**10.0**	**1.27**	-
NMSW (k=30)	0.845	0.894	0.871	0.902	0.880	**0.944**	**1.07**	**23.7**	5.85	-
NMSW (k=full)	**0.852**	**0.903**	**0.875**	**0.909**	**0.883**	**0.945**	13.0	140	63.2	53.0
Swin-UNETR										
SW (gold standard)	**0.848**	**0.897**	**0.839**	0.848	**0.846**	**0.908**	71.1	1050	69.2	15.5
Zoom-out	0.830	0.878	0.827	0.858	0.832	0.904	8.71	298	**4.17**	-
RF (k=5)	0.781	0.795	0.779	0.785	0.652	0.766	-	-	-	-
RF (k=30)	0.824	0.858	0.807	0.786	0.747	0.827	-	-	-	-
NMSW (k=5)	0.827	0.868	0.832	0.827	0.789	0.871	**0.832**	**20.2**	**1.36**	-
NMSW (k=30)	0.837	0.882	**0.847**	**0.863**	0.834	0.903	**6.07**	**94.3**	6.37	-
NMSW (k=full)	**0.846**	**0.895**	0.837	**0.863**	**0.860**	**0.920**	72.3	1140	69.2	42.0
MedNext										
SW (gold standard)	**0.860**	**0.913**	**0.898**	**0.928**	**0.909**	**0.964**	99.2	2110	88.0	17.5
Zoom-out	0.845	0.898	0.880	0.914	0.889	0.939	10.1	383	**5.21**	-
RF (k=5)	0.792	0.807	0.812	0.823	0.685	0.797	-	-	-	-
RF (k=30)	0.834	0.874	0.864	0.891	0.743	0.852	-	-	-	-
NMSW (k=5)	0.826	0.864	0.867	0.890	0.861	0.919	**1.07**	**35.8**	**1.64**	-
NMSW (k=30)	0.845	0.892	0.882	0.910	0.894	0.942	**8.37**	**189**	8.02	-
NMSW (k=full)	**0.863**	**0.916**	**0.895**	**0.927**	**0.909**	**0.956**	101	2152	88.0	44.1

of interest. However, it does not rank their importance; all foreground patches are equally likely to be selected. Also, RF ignores global prediction when local predictions are available. Zoom-out is an adaptive inference strategy that generates one patch prediction per organ. If an organ exceeds the patch size, the patch is downscaled accordingly. Both RF and Zoom-out need a global model.

We evaluate the NMSW and baselines on three popular medical image segmentation backbones: UNet [20], MedNext [21], and Swin-UNETR [5], while using UNet as the global network for computational efficiency. Each combination of the aforementioned segmentation backbones and inference techniques is tested on three multi-organ segmentation datasets: WORD [13], and Organ & Vertebrae datasets from TotalSegmentator [23]. For both NMSW and other baselines, we train for 300 epochs (500 iter/epoch) using AdamW [12] with annealed learning rate of 1e-3. We set the regularization constant $\lambda = 1e-4$. Softmax temperature τ is annealed from 2 to 0.33. The number of Top-K patches is set to 3 during training and includes 1 random patch. For baselines, 4 random patches are sampled with 2:1 fore/background ratio. patch size is set to $128 \times 128 \times 128$.

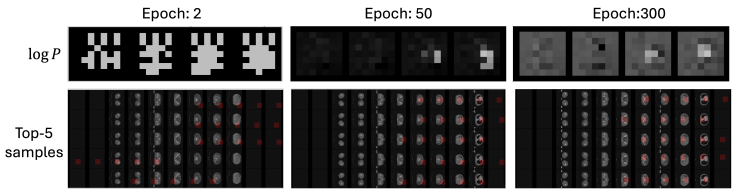

Fig. 5: Evolution of the patch sampling distribution $\mathrm{p}(\mathbf{z}|\mathbf{x}_{\mathrm{low}})$ during training.

5 Results

Trade-off between accuracy and efficiency: Table 1 compares NMSW with baselines. Although NMSW doubles model size by adding a global segmentation model, it delivers significantly better computational efficiency and competitive segmentation performance at k=30 sampled patches. While the RF is equally efficient, its patch sampling does not focus on areas where the global prediction is most deficient, yielding a smaller accuracy gain. Zoom-out is efficient, but its accuracy remains lower and static, not improving with increased patch sampling. Notably, when NMSW samples all patches (k $\approx$ 300) like SW, it even outperforms SW.

When k=30, NMSW uses about 90% fewer MACs than SW, correlating to significantly lower energy consumption [7]. Moreover, NMSW is roughly 11$\times$ faster on both CPU and GPU, with even greater speed gains as the backbone model's complexity increases. The memory consumption of NMSW during inference is approximately 10 GB higher than that of SW.

Visualizing the learned distributions: Figure 5 shows the evolution of the pmf $p(\mathbf{z}|\mathbf{x}_{\mathrm{low}})$ during training along with the top 5 sampled patches highlighted in red. Note that the images are flattened along the depth dimension for visualization, but the actual pmf and it's samples are 3D. During early training, the pmf is random and highlights all regions. By mid-training, it becomes confined to the foreground regions. Towards the end, the pmf not only focuses on the foreground but is also well-spread, thanks to a regularizer in the loss function that encourages the model to explore various regions of the image.

Ablation: We ablate the differentiable Top-K module by replacing it with RF sampling. As shown in Figure 6, Top-K saturates around $k = 30$, while RF improves linearly, saturating only when all foreground regions are sampled. We ablate the aggregation module by removing the class weight c_w_θ, forcing the model to ignore global predictions when local patches are available. Figure 6 shows that class weights generally improve the segmentation accuracy.

[7] The cost of global prediction (0.18 TMACs) and aggregation (0.07 TMACs) is negligible compared 30 local predictions (5.6 TMACs) in UNet backbone benchmark.

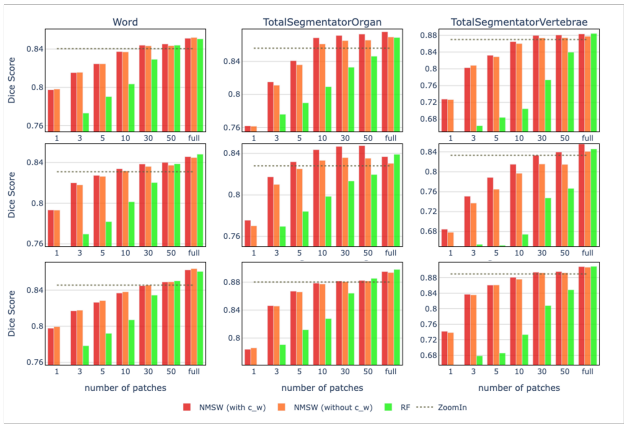

Fig. 6: Ablation of Differentiable Top-K and Aggregation blocks.

6 Conclusion & Future Works

NMSW is an innovative approach to achieving compute-efficient 3D segmentation through attention-driven patch sampling. Unlike conventional efficiency-focused methods that rely on modifying backbones—often task-specific and prone to performance trade-offs—NMSW offers a generalizable solution without sacrificing accuracy. We hope this work inspires the community to further explore dynamic sampling techniques as a promising direction for efficient 3D medical image segmentation. In future research, we aim to integrate NMSW with nnUNet to enhance segmentation accuracy.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Avesta, A., Hossain, S., Lin, M., Aboian, M., Krumholz, H.M., Aneja, S.: Comparing 3d, 2.5 d, and 2d approaches to brain image auto-segmentation. Bioengineering **10**(2), 181 (2023)
2. Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.P., Bach, F.: Learning with differentiable pertubed optimizers. Advances in neural information processing systems **33**, 9508–9519 (2020)
3. Cordonnier, J.B., Mahendran, A., Dosovitskiy, A., Weissenborn, D., Uszkoreit, J., Unterthiner, T.: Differentiable patch selection for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2351–2360 (2021)
4. Du, Y., Bai, F., Huang, T., Zhao, B.: Segvol: Universal and interactive volumetric medical image segmentation (2024), https://arxiv.org/abs/2311.13385
5. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D.: Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI brainlesion workshop. pp. 272–284. Springer (2021)

6. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)
7. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)
8. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
9. Jeon, Y.S., Yang, H., Feng, M.: Fcsn: Global context aware segmentation by learning the fourier coefficients of objects in medical images. IEEE Journal of Biomedical and Health Informatics **28**(3), 1195–1206 (2022)
10. Jeon, Y.S., Yang, H., Fu, H., Feng, M.: Teaching ai the anatomy behind the scan: Addressing anatomical flaws in medical image segmentation with learnable prior. arXiv preprint arXiv:2403.18878 (2024)
11. Kool, W., Van Hoof, H., Welling, M.: Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In: International Conference on Machine Learning. pp. 3499–3508. PMLR (2019)
12. Loshchilov, I.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
13. Luo, X., Liao, W., Xiao, J., Chen, J., Song, T., Zhang, X., Li, K., Metaxas, D.N., Wang, G., Zhang, S.: Word: A large scale dataset, benchmark and clinical applicable study for abdominal organ segmentation from ct image. Medical Image Analysis **82**, 102642 (Nov 2022). https://doi.org/10.1016/j.media.2022.102642, http://dx.doi.org/10.1016/j.media.2022.102642
14. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712 (2016)
15. Man, Y., Huang, Y., Feng, J., Li, X., Wu, F.: Deep q learning driven ct pancreas segmentation with geometry-aware u-net. IEEE transactions on medical imaging **38**(8), 1971–1980 (2019)
16. Mnih, V.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
17. Perera, S., Navard, P., Yilmaz, A.: Segformer3d: an efficient transformer for 3d medical image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4981–4988 (2024)
18. Qin, D., Bu, J.J., Liu, Z., Shen, X., Zhou, S., Gu, J.J., Wang, Z.H., Wu, L., Dai, H.F.: Efficient medical image segmentation based on knowledge distillation. IEEE Transactions on Medical Imaging **40**(12), 3820–3831 (2021)
19. Recasens, A., Kellnhofer, P., Stent, S., Matusik, W., Torralba, A.: Learning to zoom: a saliency-based sampling layer for neural networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 51–66 (2018)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. pp. 234–241. Springer (2015)
21. Roy, S., Koehler, G., Ulrich, C., Baumgartner, M., Petersen, J., Isensee, F., Jaeger, P.F., Maier-Hein, K.H.: Mednext: transformer-driven scaling of convnets for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 405–415. Springer (2023)
22. Wang, H., Lin, L., Hu, H., Chen, Q., Li, Y., Iwamoto, Y., Han, X.H., Chen, Y.W., Tong, R.: Patch-free 3d medical image segmentation driven by super-resolution

technique and self-supervised guidance. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24. pp. 131–141. Springer (2021)
23. Wasserthal, J., Breit, H.C., Meyer, M.T., Pradella, M., Hinck, D., Sauter, A.W., Heye, T., Boll, D.T., Cyriac, J., Yang, S., et al.: Totalsegmentator: robust segmentation of 104 anatomic structures in ct images. Radiology: Artificial Intelligence **5**(5) (2023)
24. Zeng, G., Zheng, G.: Holistic decomposition convolution for effective semantic segmentation of medical volume images. Medical image analysis **57**, 149–164 (2019)