Efficient Fine-Tuning of CNN-based Foundation Models for Segmentation in 3D Medical Images

Mees Hudepohl
AIM Harvard, US
Maastricht University, NL
mhudepohl@bwh.harvard.edu

Suraj Pai
AIM Harvard, US
Maastricht University, NL
bspai@bwh.harvard.edu

Ibrahim Hadzic
AIM Harvard, US
Maastricht University, NL
ihadzic@bwh.harvard.edu

Heysem Kaya
Utrecht University, NL
h.kaya@uu.nl

Hugo Aerts
AIM Harvard, US
Maastricht University, NL
haerts@bwh.harvard.edu

Abstract

Medical imaging techniques like Computed Tomography (CT) are crucial for disease detection and treatment, with semantic segmentation being essential for accurate analysis. Despite the potential of deep learning models, particularly Convolutional Neural Networks (CNNs), for automated segmentation, the limited availability of labeled data in medical imaging remains an obstacle. To address this problem, foundation models have been introduced, which require fine-tuning to adapt to specific tasks. However, state-of-the-art methods like full fine-tuning are storage-intensive and prone to forgetting and overfitting. As a more efficient alternative, Parameter-Efficient Fine-Tuning (PEFT) techniques have been developed. Nevertheless, most PEFT research has been concentrated on transformer-based models applied to language or 2D natural images, leaving a gap in the application of these techniques to CNN-based models for 3D medical imaging. This study addresses the gap by applying the PEFT technique ConvAdapter to a supervised SegResNet, a CNN-based segmentation model, for segmenting organs in 3D CT images. The goal is to enhance performance while minimizing the number of tunable parameters. We demonstrate that integrating ConvAdapter within SegResNet achieves an effective balance between performance and parameter efficiency, yielding a Mean Dice score of 0.84 on the test set while only tuning 0.7M parameters - less than 15% of the total model parameters. ConvAdapter maintains performance trends similar to full fine-tuning and shows promising generalization across diverse datasets, even outperforming full fine-tuning on MR data. These findings highlight the potential of PEFT techniques in improving the efficiency of fine-tuning CNN models for medical imaging, particularly for complex tasks like 3D organ segmentation. By refining these techniques and exploring their integration with self-supervised foundation models, they hold promise for developing even more adaptable and efficient models.

1 Introduction

Medical imaging techniques like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are crucial for detecting and diagnosing diseases [37], but analyzing 3D medical images, particularly segmenting tissues or anomalies, remains challenging [13]. Deep Learning (DL)-based automated segmentation methods have shown promise [21], yet they are often hindered by the

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

scarcity of labeled data within the medical imaging domain [19]. To address this, researchers are developing foundation models pre-trained on large datasets, which can be fine-tuned for specific tasks like segmentation with minimal labeled data [39, 3]. So far, foundation models have shown great potential in learning feature representations across various medical imaging modalities, beneficial for downstream tasks like classification and segmentation [34, 14].

Foundation models, pre-trained through supervised or self-supervised learning, require adaptation to specific tasks via fine-tuning techniques [3, 7], commonly full fine-tuning and linear probing. Full fine-tuning, which adjusts all model parameters, is computationally demanding as it creates entirely new models for each specific task and can risk overfitting as adjustments to all parameters can lead to distortion or forgetting of the original features learned during pre-training [27, 20]. In contrast, linear probing, which only updates the model's final layers, adapts less to new tasks and may result in suboptimal performance [25].

To balance efficiency and accuracy, Parameter-Efficient Fine-Tuning (PEFT) techniques are being explored [7, 9]. PEFT techniques focus on training and optimizing only a small set of parameters, which could either be a subset of the existing model parameters or a set of newly added parameters [25]. While most PEFT techniques are being developed particularly for Transformer-based models [9, 7, 38, 26], Convolutional Neural Networks (CNNs) continue to play a critical role in 3D medical imaging [5, 31, 10, 16]. The development of PEFT techniques for CNN-based foundation models is still in its early stages, and most efforts are focused on models pre-trained on 2D natural images.

In this study, we aim to adapt and optimize PEFT for pre-trained 3D CNN-based foundation models within medical imaging, specifically focusing on semantic segmentation in CT. The PEFT technique applied in our study is ConvAdapter, an additive PEFT technique specifically designed for CNNs [15, 18, 6]. By systematically applying and customizing ConvAdapter to accommodate the characteristics of SegResNet [29], a state-of-the-art CNN-based segmentation model, we seek to enhance the efficiency of fine-tuning on semantic segmentation tasks. We specifically focus on organ segmentation, a critical task in computer-assisted diagnostic systems, biomarker measurements, and radiation therapy planning [11, 8], necessitating precise delineation of anatomical structures.

By exploring the synergy between the SegResNet architecture and ConvAdapter, we expect to make advancements in medical image analysis, contributing to the development of more accurate and efficient organ segmentation algorithms, addressing critical clinical needs, and improving disease detection and treatment.

2 Methods

2.1 Foundation Model

This study includes a CNN-based foundation model for integrating the PEFT technique ConvAdapter. This foundation model is a SegResNet with 4.7 million parameters and is trained using the SuPreM [24] framework on a dataset of 2100 labeled 3D CT images from the AbdomenAtlas 1.1 multi-organ dataset [23] containing segmentations of 25 structures, including 16 abdominal organs (esophagus, stomach, duodenum, intestine, colon, rectum, liver, gall bladder, spleen, pancreas, left kidney, right kidney, left adrenal gland, right adrenal gland, bladder, prostate), 2 thorax organs (left lung, right lung), 5 vascular structures (aorta, celiac trunk, postcava, portal & splenic vein, hepatic vessel), and 2 skeletal structures (left and right femur). Due to the labeled dataset, the SegResNet has learned image features through supervised learning.

2.2 Downstream Data

For the segmentation downstream task, the open source TotalSegmentator v2 dataset [35] is utilized. This dataset contains 1228 3D CT images segmented with 117 anatomical structures. The anatomical structures are divided into five classes: cardiac, muscles, organs, ribs, and vertebrae. The initial focus is on segmenting the organs class, which contains 17 organ subclasses. The ground truth organ segmentations, as well as the predicted segmentations, are represented in tensor format. The input tensor is $X \in \mathbb{R}^{W \times H \times D}$, where W and H are the width and height of each slice, and D is the depth. The output tensor is $Y \in \{0, 1\}^{W \times H \times D \times K}$, with K being the number of classes (K = 17 for organ class segmentation). Each element in the output tensor indicates the presence of a class at each spatial location in the 3D volume.

2.2.1 Downstream Data Pre-processing

The dataset of 1228 images is randomly split into training, validation and testing subsets, with a selected distribution of 70%, 20%, and 10% respectively. This results in 859 images for training, 245 images for validation, and 124 images for testing.

To be fed into the employed model for the organ segmentation task, the images and corresponding segmentations with their class labels are converted to dictionaries. Prior to model input, a series of data transformations is applied, sourced from the MONAI library [4]. These transformations are provided in Supplementary **A**.

2.3 Baseline

In this study, a foundation model from SuPreM, as described in Section 2.1 is fully fine-tuned as baseline, where all model parameters are adjusted during training to meet the specific requirements of the organ segmentation task. This baseline serves as a reference to compare the performance of ConvAdapter. The hyperparameters for training the baseline and ConvAdapter can be found in Supplementary **B**.

2.4 ConvAdapter

Chen et al. proposed ConvAdapter, a PEFT technique for CNNs that enhances transferability and parameter efficiency without tuning backbone parameters [6]. Unlike other PEFT techniques, ConvAdapter preserves spatial feature size, important for tasks like segmentation, using a bottleneck structure with two convolutional layers and a non-linearity. The first layer reduces channels, while the second restores them, resembling the receptive fields of the backbone. Depth-wise convolutions are used to reduce parameters. The architecture of ConvAdapter is shown in Figure 1.

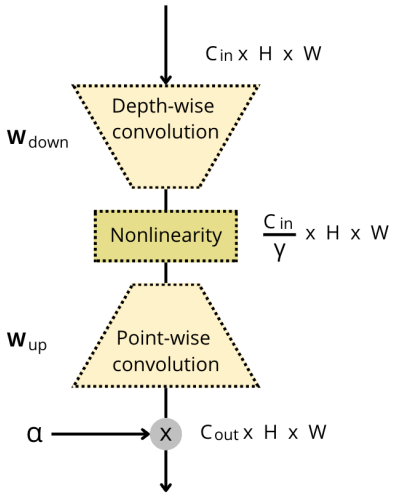

Figure 1: **ConvAdapter bottleneck architecture.** Adapted from [6]. The first convolutional layer downsamples the channels with a kernel size matching the adapted blocks, while the second restores the original channels. α and γ are hyper-parameters to tune.

The input feature map has dimensions $C_{in} \times H \times W$, and the output is $C_{out} \times H \times W$, where C_{in} are the input channels, C_{out} the output channels, and H and W are the height and width of the feature maps, respectively. The depth-wise convolution weight W_{down} is sized $\frac{C_{in}}{\gamma} \times \gamma \times K \times K$, and the point-wise convolution weight W_{up} is $C_{out} \times \frac{C_{in}}{\gamma} \times 1 \times 1$, where γ is the compression factor for down-sampling the channel dimension.

3 Findings

3.1 ConvAdapter Implementation

ConvAdapter is tested in both the encoder and decoder of SegResNet, as well as solely in the encoder. The best performance is achieved with ConvAdapter in both the encoder and decoder, placed in parallel with the entire ResBlocks. Here, all pre-trained weights are kept frozen, while ConvAdapter is being updated. As for the hyper-parameters α and γ, both are set to 1 following the results of a prior sensitivity analysis conducted by the creators of ConvAdapter. This setup is used for further evaluation. The performance of other configurations and training setups tested is provided in Supplementary **C**.

3.2 ConvAdapter Evaluation

The implementation of ConvAdapter into the SegResNet is compared with the baseline in terms of the trade-off between performance and parameter efficiency, the performance on different downstream dataset sizes, Out Of Distribution (OOD) data, computational costs, and interpretability.

3.2.1 Performance - Parameter Efficiency Trade-off

Performance is assessed using the Mean Dice score on the test set, while parameter efficiency is measured by the number of parameters. We compare ConvAdapter with full fine-tuning (FFT) and decoder tuning, a segmentation oriented linear probing equivalent [17]. We use the best performing ConvAdapter configuration as analysed in Section 3.1, with only ConvAdapter being tuned on the downstream data while the pre-trained weights remain frozen. Full fine-tuning tunes all model parameters, while decoder tuning focuses solely on tuning the decoder parameters on the downstream data. Figure 2 illustrates the performance and parameter efficiency trade-off of these methods.

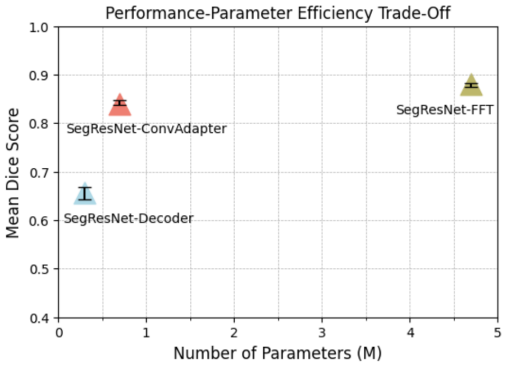

Figure 2: **Performance of ConvAdapter on test set.** The performance of ConvAdapter inserted in parallel with all ResBlocks within the encoder and decoder in SegResNet on the test set is shown and compared with full fine-tuning (SegResNet-FFT) and decoder tuning (SegResNet-Decoder). The 95% CI of the estimates is depicted with error bars. Wilcoxon signed-rank tests revealed statistically significant differences ($p < 0.001$) between ConvAdapter and the other methods across the organ classes within the n=124 test samples.

3.2.2 Downstream Dataset Sizes

ConvAdapter's effectiveness on different dataset sizes of the TotalSegmentator CT data is assessed. The full dataset of 1228 images (859 for training, 245 for validation, and 124 for testing) is used as the 100% set. Fine-tuning is also done on 50% and 10% of the training/validation images, maintaining an 80/20 training-validation split. The test set is the same for all experiments. The performance of ConvAdapter on the various dataset sizes is compared against full fine-tuning on the same dataset sizes. The results are shown in Figure 3.

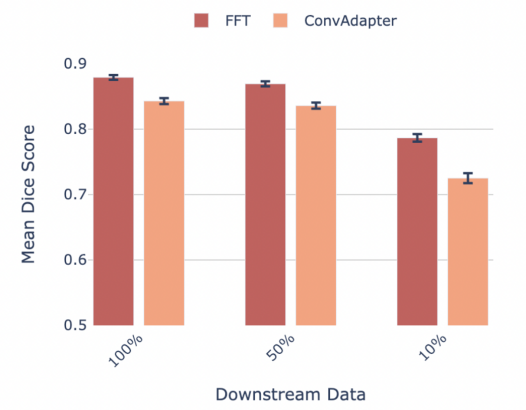

Figure 3: **Performance of ConvAdapter in SegResNet on test set with various dataset sizes.** The performance of ConvAdapter in the encoder and decoder of the SegResNet is shown for inference on the test set after being trained on various sizes of the training and validation downstream data, and compared with full fine-tuning. The 95% CI of the estimates is depicted with error bars. Wilcoxon signed-rank tests revealed statistically significant differences ($p < 0.001$) between ConvAdapter and the full SegResNet tuned on different dataset sizes for the organ classes within the n=124 test samples.

3.2.3 Out Of Distribution(OOD) Data

To determine the robustness of Convadapter, its performance on OOD data is determined. The first OOD data contains the vertebrae class of the TotalSegmentator CT dataset. The TotalSegmentator v2 dataset, used in this study, includes four additional classes beyond organs: cardiac, muscles, ribs, and vertebrae. Since the SuPreM pre-trained models were trained only on organs (AbdomenAtlas 1.0), these four classes are considered out-of-distribution. For evaluating ConvAdapter's robustness, we use 100% of the TotalSegmentator 3D CT images with vertebrae segmentations, totaling 1228 images.

Second, the PEFT technique is trained and tested on MR images and organ segmentations from the TotalSegmentator MR dataset [36]. This dataset includes 298 MR scans with segmentations for five classes: organs, bones, muscles, vessels, and tissue types. Since the foundation model is pre-trained only on organs, the MR dataset's organ class is used and presents out-of-distribution data due to differences in imaging modality [1] and a more comprehensive organ classification (29 subclasses). After excluding images lacking subclass segmentations, the dataset for evaluation comprises 235 images (164 for training, 47 for validation, and 24 for testing).

Figure 4.A and 4.B show the performance of ConvAdapter on segmenting the vertebrae class and the MR organ class, respectively. Full fine-tuning seems to perform better when differences are higher-level abstractions such as concepts of different organs. Full fine-tuning, which allows the entire model to adapt, can capture the specific anatomical features of vertebrae [22]. In contrast, ConvAdapter, which only adjusts additional parameters while keeping the pre-trained features fixed, may have limited capacity to fully adapt to the nuances of vertebrae segmentation.

When moving to a domain where differences exist in lower-level abstractions that are usually learned in the first few layers of the network, such as contrast differences, edges, textures, intensity variations, we speculate that PEFT surpasses full fine-tuning. PEFT techniques like ConvAdapter enhance generalization by retaining these important learned features [2]. This is verified by adaptation on the MRI use-case where large differences in such abstractions exist. In the MRI use-cases, high-level abstractions remain similar while lower-level abstractions are relatively different. ConvAdapter offering better performance in lower-level abstractions might offer strong practical benefits such as adaptation across different scanner types and acquisitions, MRI sequences, low-field vs high-field MRI.

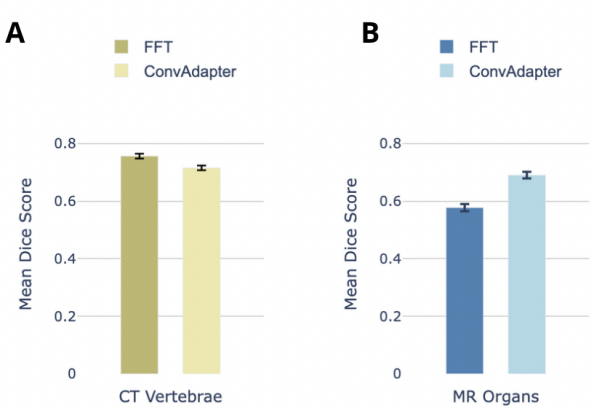

Figure 4: **Performance of ConvAdapter in SegResNet on Out Of Distribution (OOD) data.** After being tuned on the Vertebrae class of the TotalSegmentator CT dataset (A) and the Organ class of the TotalSegmentator MR dataset (B), inference on the corresponding test sets is performed by both ConvAdapter in the encoder and decoder of the SegResNet and the fully tuned SegResNet. The 95% CI of the estimates is depicted with error bars. Wilcoxon signed-rank tests revealed statistically significant differences ($p < 0.001$) between SegResNet with ConvAdapter and the fully tuned SegResNet on all CT vertebrae classes and all MR organ classes - except lung upper lobe left - within the n=124 test samples.

3.2.4 Computational Costs

To compare ConvAdapter with full fine-tuning of the SegResNet in terms of computational costs, the training time, inference time, and GPU memory are determined for both. The results in Table 1 reveal that ConvAdapter offers nearly identical training and inference times, with a modest increase in GPU memory utilization.

The similar training time could be due to maintaining the full computational graph for gradient computation across the entire model, despite fewer tunable parameters. Even if only a subset of parameters is being tuned, gradients must be computed for the entire model to update the trainable parameters correctly [28]. The similar inference time results from forwarding through both the backbone and ConvAdapter, despite only tuning ConvAdapter [6]. The slight increase in GPU memory is likely due to caching intermediate activations for gradient calculation in ConvAdapter, which still requires backpropagation through the entire pre-trained model [32].

	Computational Costs	
	FFT	ConvAda
Training Time (s/volume)	2.10	2.11
Inference Time (s/volume)	0.12	0.14
GPU Memory (GB)	11.01	11.75

Table 1: **Computational costs of ConvAdapter in SegResNet.** Tuning ConvAdapter in the SegResNet on the downstream data is compared with full fine-tuning the SegResNet.

3.2.5 Interpretability

Grad-CAM Grad-CAM is used to generate visual explanations of SegResNet with ConvAdapter, highlighting relevant regions for segmenting specific organs. It is applied to the final convolutional layer, the second convolution of the last upsampling layer, and the convolution within the ConvAdapter that runs parallel to it. These convolutions are summed as input to the final layer, revealing

ConvAdapter's impact on the output. The resulting maps are shown in Figure 5. These show that the last upsampling layer of SegResNet and the last ConvAdapter layer complement each other, indicating their synergy in enhancing segmentation. This alignment helps clarify their combined impact on the model's decision-making.

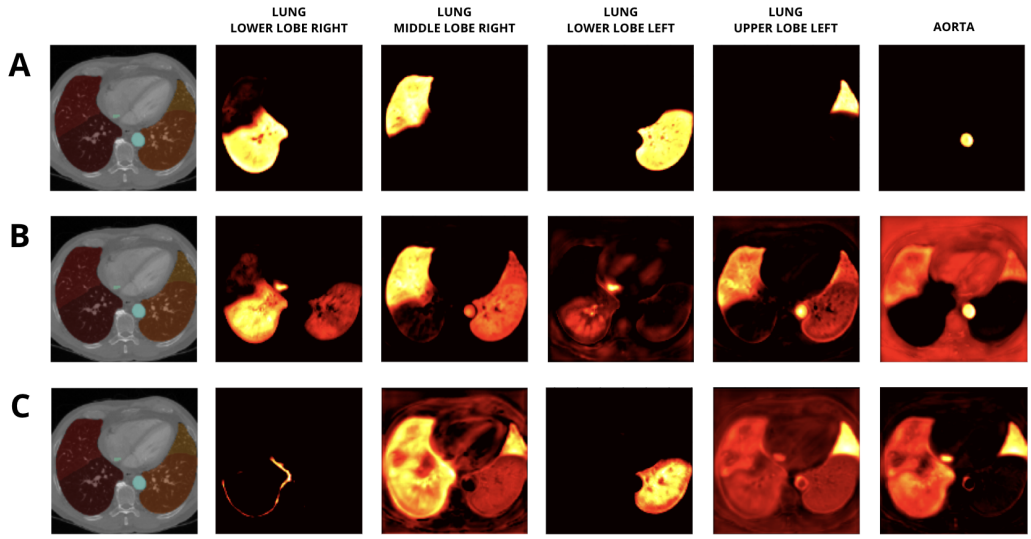

Figure 5: **Grad-CAM on the final layers of SegResNet with ConvAdapter.** (A) Grad-CAM on final convolutional layer: Highlights the regions most influential for the final segmentation decision. (B) Grad-CAM on last convolutional layer in ConvAdapter parallel to last ResBlock: Shows the areas emphasized by ConvAdapter, indicating its contribution to the segmentation process. (C) Grad-CAM on last convolutional layer in last ResBlock: Shows the regions captured by the standard ResBlock. First column depicts the ground truth segmentations.

ProtoSeg To investigate the contribution to the segmentation process of SegResNet's convolutional layers, including those within ConvAdapter, ProtoSeg is used. ProtoSeg calculates prototypes for object and background regions based on the average values in intermediate features of a network, using segmentation outputs from the neural network as guidance. These prototypes are then used to segment all the pixels or voxels on the feature map, creating a binary Segmentation Ability Map (SAM). It then evaluates the segmentation performance by measuring the Segmentation Ability (SA) score, which is the Dice between the binary feature segmentation map and the ground-truth. This score assesses how effectively different layers contribute to segmentation [12]. In Figure 6, the scores of SegResNet with ConvAdapter are shown. Particularly in the final layers, ConvAdapter shows a great contribution.

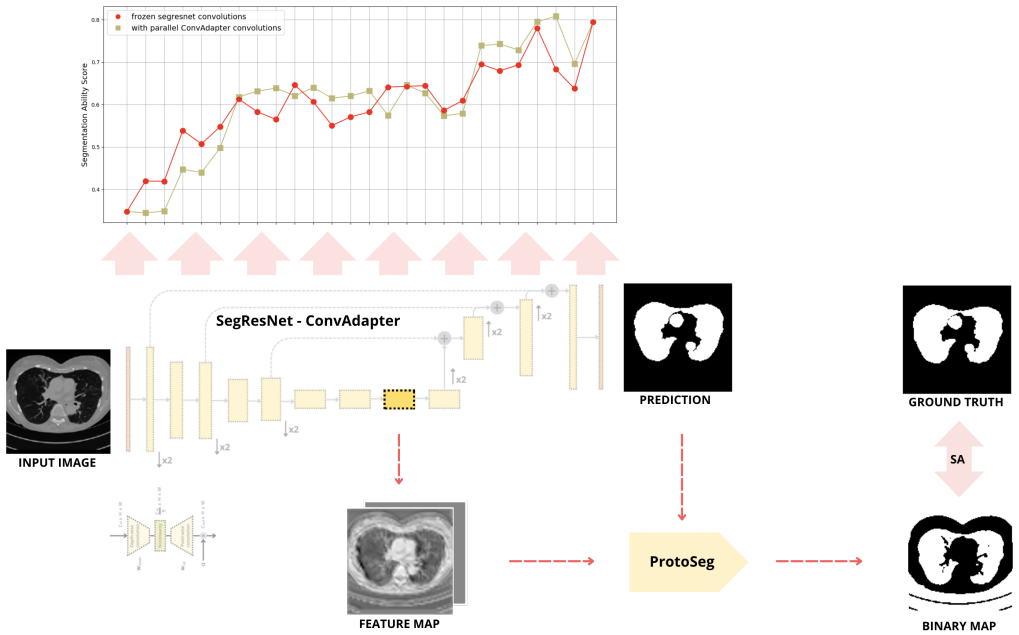

Figure 6: **ProtoSeg on the convolutional layers in SegResNet with ConvAdapter.** The Segmentation Ability (SA) score of all convolutional layers in the SegResNet with ConvAdapter are plotted in the graph. As ConvAdapter is placed in parallel with the ResBlocks in the SegResNet, the scores of the convolutional layers within the ResBlocks and the scores of ConvAdapter's convolutional layers are also displayed in parallel in the graph. The convolutions within the ResBlocks contain frozen pre-trained weights, while the convolutions within ConvAdapter are trained on the downstream data. Notably, the images shown in this figure are slices of the original 3D images for the sake of visualization.

References

[1] Edwin JR van Beek and Eric A Hoffman. "Functional imaging: CT and MRI". In: *Clinics in chest medicine* 29.1 (2008), pp. 195–216.

[2] Dan Biderman et al. *LoRA Learns Less and Forgets Less*. 2024. arXiv: 2405.09673 [cs.LG]. URL: https://arxiv.org/abs/2405.09673.

[3] Rishi Bommasani et al. "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258* (2021).

[4] M. Jorge Cardoso et al. "MONAI: An open-source framework for deep learning in healthcare". In: (Nov. 2022). DOI: https://doi.org/10.48550/arXiv.2211.02701.

[5] Pedro Celard et al. "A survey on deep learning applied to medical images: from simple artificial neural networks to generative models". In: *Neural Computing and Applications* 35.3 (2023), pp. 2291–2323.

[6] Hao Chen et al. "Conv-adapter: Exploring parameter efficient transfer learning for convnets". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 1551–1561.

[7] Ning Ding et al. "Parameter-efficient fine-tuning of large-scale pre-trained language models". In: *Nature Machine Intelligence* 5.3 (2023), pp. 220–235.

[8] Yabo Fu et al. "A review of deep learning based methods for medical image multi-organ segmentation". In: *Physica Medica* 85 (2021), pp. 107–122.

[9] Zihao Fu et al. "On the effectiveness of parameter-efficient fine-tuning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 11. 2023, pp. 12799–12807.

[10] C Ghandour, Walid El-Shafai, and S El-Rabaie. "Medical image enhancement algorithms using deep learning-based convolutional neural network". In: *Journal of Optics* 52.4 (2023), pp. 1931–1941.

[11] Eli Gibson et al. "Automatic multi-organ segmentation on abdominal CT with dense V-networks". In: *IEEE transactions on medical imaging* 37.8 (2018), pp. 1822–1834.

[12] Sheng He et al. "Segmentation ability map: Interpret deep features for medical image segmentation". In: *Medical image analysis* 84 (2023), p. 102726.

[13] Mohammad Hesam Hesamian et al. "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges". In: *Journal of Digital Imaging* 32 (2019), pp. 582–596. URL: https://api.semanticscholar.org/CorpusID:169033851.

[14] Mohammad Reza Hosseinzadeh Taher, Michael B Gotway, and Jianming Liang. "Towards foundation models learned from anatomy in medical imaging via self-supervision". In: *MICCAI Workshop on Domain Adaptation and Representation Transfer*. Springer. 2023, pp. 94–104.

[15] Neil Houlsby et al. "Parameter-efficient transfer learning for NLP". In: *International conference on machine learning*. PMLR. 2019, pp. 2790–2799.

[16] Fabian Isensee et al. *nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation*. 2024. arXiv: 2404.09556 [cs.CV]. URL: https://arxiv.org/abs/2404.09556.

[17] Zhanghexuan Ji et al. "Continual segment: Towards a single, unified and non-forgetting continual segmentation model of 143 whole-body organs in ct scans". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 21140–21151.

[18] Menglin Jia et al. "Visual Prompt Tuning". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Berlin, Heidelberg, 2022, pp. 709–727.

[19] Rushi Jiao et al. "Learning with limited annotations: a survey on deep semi-supervised learning for medical image segmentation". In: *Computers in Biology and Medicine* (2023), p. 107840.

[20] Ananya Kumar et al. "Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=UYneFzXSJWh.

[21] Leon Lenchik et al. "Automated segmentation of tissues using CT and MRI: a systematic review". In: *Academic radiology* 26.12 (2019), pp. 1695–1706.

[22] Nikolas Lessmann et al. "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification". In: *Medical Image Analysis* 53 (Apr. 2019), pp. 142–155. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.02.005. URL: http://dx.doi.org/10.1016/j.media.2019.02.005.

[23] W. Li et al. "AbdomenAtlas: A large-scale, detailed-annotated, multi-center dataset for efficient transfer learning and open algorithmic benchmarking". In: *Medical Image Analysis* 97 (2024), p. 103285. DOI: 10.1016/j.media.2024.103285.

[24] Wenxuan Li, Alan Yuille, and Zongwei Zhou. "How Well Do Supervised Models Transfer to 3D Image Segmentation?" In: *The Twelfth International Conference on Learning Representations*. 2024.

[25] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. "Scaling down to scale up: A guide to parameter-efficient fine-tuning". In: *arXiv preprint arXiv:2303.15647* (2023).

[26] Chenyu Lian et al. "Less Could Be Better: Parameter-efficient Fine-tuning Advances Medical Vision Foundation Models". In: *Medical Imaging with Deep Learning*. 2024. URL: https://openreview.net/forum?id=GVxHxL3HIp.

[27] Dongze Lian et al. "Scaling & shifting your features: A new baseline for efficient model tuning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 109–123.

[28] Yitao Liu, Chenxin An, and Xipeng Qiu. "Y-tuning: An efficient tuning paradigm for large-scale pre-trained models via label representation learning". In: *Frontiers of Computer Science* 18.4 (2024), p. 184320.

[29] Andriy Myronenko. "3D MRI brain tumor segmentation using autoencoder regularization". In: *BrainLes@MICCAI*. 2018. URL: https://api.semanticscholar.org/CorpusID:53104235.

[30] Sajad Norouzi and M Ebrahimi. "A survey on proposed methods to address Adam optimizer deficiencies". In: *Department of Electrical and Computer Engineering, University of Toronto* (2019).

[31] Ahmad Waleed Salehi et al. "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope". In: *Sustainability* 15.7 (2023), p. 5930.

[32] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. "LST: ladder side-tuning for parameter and memory efficient transfer learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/54801e196796134a2b0ae5e8adef502f-Abstract-Conference.html.

[33] Aditya Thakur, Harish Chauhan, and Nikunj Gupta. "Efficient ResNets: Residual Network Design". In: *arXiv preprint arXiv:2306.12100* (2023).

[34] Guotai Wang et al. "Mis-fm: 3d medical image segmentation using foundation models pre-trained on a large-scale unannotated dataset". In: *arXiv preprint arXiv:2306.16925* (2023).

[35] Jakob Wasserthal. *Dataset with segmentations of 117 important anatomical structures in 1228 CT images (2.0.1)*. Zenodo. Version 2.0.1. 2023. DOI: 10.5281/zenodo.10047292.

[36] Jakob Wasserthal and Taha Akinci D'Antonoli. *TotalSegmentator MRI dataset: 298 MRI images with segmentations for 56 anatomical regions*. Version 1.0.0. 2024. DOI: 10.5281/zenodo.11367005. URL: https://doi.org/10.5281/zenodo.11367005.

[37] Guorong Wu and Heung-Il Suk. "Deep Learning in Medical Image Analysis". In: *Annual review of biomedical engineering* 19 (Mar. 2017). DOI: 10.1146/annurev-bioeng-071516-044442.

[38] Yi Xin et al. "Parameter-Efficient Fine-Tuning for Pre-Trained Vision Models: A Survey". In: *arXiv preprint arXiv:2402.02242* (2024).

[39] Shaoting Zhang and Dimitris Metaxas. "On the challenges and perspectives of foundation models for medical image analysis". In: *Medical Image Analysis* (2023), p. 102996.

Supplementary Material

A. Downstream Data Pre-processing

A series of data transformations from the MONAI library is applied to the downstream data [4]. For the training set the transformations include: AddChanneld, Orientationd, CropForegroundd, Spacingd, ScaleIntensityRanged, ResizeWithPadOrCropd, RandRotate90d, RandShiftIntensityd, and ToTensord. For the validation and test set these transformations include: AddChanneld, Orientationd, CropForegroundd, Spacingd, ScaleIntensityRanged, ResizeWithPadOrCropd, and ToTensord.

B. Hyperparameters

Due to GPU memory constraints, the batch size for each experiment is set to 2. As an optimizer, AdamW is selected due to handling weight decay decoupled from the learning rate, which helps in better regularization and preventing overfitting [30]. The learning rate for AdamW is set to 1e-4 and the weight decay to 1e-5 based on extensive hyperparameter ablations proposed by the MONAI framework [4]. Additionally, the learning rate is dynamically adjusted using the CosineAnnealingLR scheduler to avoid local minima and getting better convergence [33].

C. ConvAdapter Configurations

ConvAdapter was tested in both the encoder and decoder of the SegResNet, or solely in the encoder. For the combined setup, all pre-trained weights were frozen during training, with only the adapters being updated. Configurations included ConvAdapter in parallel with the first convolution (ConvInit) and the entire ResBlocks, only with the ResBlocks, and in parallel with the second convolution within the ResBlocks. For the encoder-only setup, ConvAdapter was used in parallel only with the ResBlocks, with the encoder weights frozen while either the adapters and decoder or just the adapters were updated. In Table 2, the results of integrating ConvAdapter into the SegResNet are shown.

	Full Fine Tuning	ConvAdapter Fine Tuning				
		Encoder + Decoder			Encoder (+ full decoder)	Encoder (no full decoder)
		ConvInit + ResBlock Parallel	ResBlock Parallel	ResBlock Conv2 Parallel	ResBlock Parallel	ResBlock Parallel
# Param (M)	4.7 (100%)	0.75 (15.9%)	0.7 (14.9%)	0.7 (14.9%)	0.9 (19.2%)	0.65 (13.8%)
Val Mean Dice	**0.85**	0.77	<u>0.82</u>	0.72	0.70	0.50

Table 2: **Comparison of Full Fine-Tuning and Fine-Tuning with ConvAdapter integrated in the SegResNet.** *ConvAdapter was integrated in either both the encoder and decoder, or solely the encoder of the SegResNet. Integrated in the encoder and decoder, the adapters were placed in parallel with the first convolution (ConvInit) and with the entire ResBlocks, with only the entire ResBlocks, or with the second convolutional layers within the ResBlocks. In all setups, the pre-trained weights were kept frozen during training, while the adapters were being updated. Integrated in solely the encoder of the SegResNet, the adapters were placed in parallel with the entire ResBlocks. Here, the pre-trained weights in the encoder were kept frozen during training, and the adapters together with the full decoder were being tuned. The adapters in the encoder were also tuned without the full decoder. The best performing approach is highlighted in bold, and second best is underlined.*

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We hope that following discussions and feedback from NeurIPS, to submit our paper to a journal. During this point, we will clean up and organize our codebase so it is effectively useful to other researchers.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.