Self-Distillation Representation Learning and Parameter Efficient Fine-Tuning for Pretrained Models in Multimodal 3D Medical Imaging

Tony Xu[1 [0000−0002−9825−1005]], Anne L. Martel[1,2 [0000−0003−1375−5501]], and Maged Goubran[1,2,3,4 [0000−0001−5880−0818]]

[1] Department of Medical Biophysics, University of Toronto, Toronto, Canada
[2] Physical Sciences Platform, Sunnybrook Research Institute, Toronto, Canada
[3] Hurvitz Brain Sciences, Sunnybrook Health Sciences Centre, Toronto, Canada
[4] Harquail Centre for Neuromodulation, Sunnybrook Health Sciences Centre, Toronto, Canada
tonylt.xu@mail.utoronto.ca

Abstract. Applying deep learning (DL) to medical imaging generally requires large amounts of expert-annotated data. Foundation models, that are pretrained on huge unlabeled datasets offer a promising way to reduce this reliance. While self-supervised learning (SSL) has advanced foundation model development for 2D natural and medical images, extending these methods to 3D medical imaging remains computationally challenging and often constrained by limited pretraining data. In this work, we adapt the 3DINO framework—an extension of DINOv2 to volumetric inputs—to the MICCAI-FLARE25 Challenge Task 4, leveraging 20,000 unlabeled CT and MRI scans for pretraining. To efficiently transfer learned representations to diverse tasks while avoiding overfitting, we employ parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA). Our results demonstrate that combining 3DINO pretraining with LoRA improves performance across segmentation, classification, survival prediction, and regression. These findings highlight the potential of SSL-based pretrained models to enable more label-efficient training for diverse 3D medical imaging applications.

Keywords: Self-supervised learning · Medical imaging · Parameter-efficient fine-tuning

1 Introduction

Enlisting the support of trained medical professionals to create detailed annotations for training deep learning (DL) models for medical imaging applications is both expensive and time-consuming. In 2D natural images, numerous works have employed self-supervised learning (SSL) to great success in scaling to larger and more diverse pre-training datasets and creating image representations that are generalizable to many downstream tasks [3,1,7,20,9]. These methods have been shown to be highly successful in reducing the amount of annotations required

to train DL models for 2D natural images, and more recently have benefited 2D medical image applications as well [2,16]. The ability of such pretrained models to learn from vast unannotated datasets in both computer vision and natural language processing have prompted the creation of the term "foundation models": models that can generalize to huge amounts of downstream tasks without significant amounts of labeled data or tuning.

In the domain of SSL applied to 3D medical images, Taleb et al. [15] introduced a suite of self-supervised tasks, including rotation prediction and shuffle. Another related study by Chen et al. [4] investigated the efficacy of masked image modeling for creating representations of medical images. This generative SSL technique involves masking patch regions in an input image, and tasking a Vision Transformer (ViT) [6] model to predict the masked locations. While these methods introduced in the literature show that SSL can be an effective tool for improving the label-efficiency of training ML models on medical imaging datasets, they performed SSL pre-training on relatively smaller datasets that were aligned to the fine-tuning task in both organ and modality. This limits the scalability and generalizability of their methods, requiring in-domain pre-training datasets, and a separate model to be trained for each downstream task.

While existing SSL methods for images are excellent for 2D representation learning, scaling them to 3D is prohibitively computationally expensive due to the large batch sizes needed to train effectively. The recently proposed DINOv2 SSL methodology [14] provides numerous improvements in computational efficiency and training stability relative to its counterparts, which motivated the creation of 3DINO [17], a work adapting DINOv2 to 3D medical imaging inputs. In this work, we apply the 3DINO framework to the MICCAI-FLARE25 Challenge Task 4: Foundation Models for 3D CT and MRI. This challenge aims to develop and tune foundation models for *volumetric* medical imaging modalities using 10,000 unlabeled CT scans and 10,000 unlabeled MRI scans. While 3DINO appeared to generalize well to the challenge dataset, we also expect the scale of pretraining to be suboptimal, especially given that 2D datasets easily surpass 1,000,000 unlabeled pretraining images. Thus, we enhance and adjust our pretrained model weights for downstream tasks while maintaining generalizability using parameter-efficient fine-tuning methods. Specifically, in this work we use the well known Low-Rank Adaptation (LoRA) method.

2 Method

Our overall method involves first using 3DINO to pretrain a ViT-Large [6] on the *pooled* CT and MRI unlabeled datasets (approx. 20,000 images total). Then, using LoRA to adapt pretrained model weights, we perform downstream fine-tuning on all challenge tasks. A summary of our method can be seen in Figure 1.

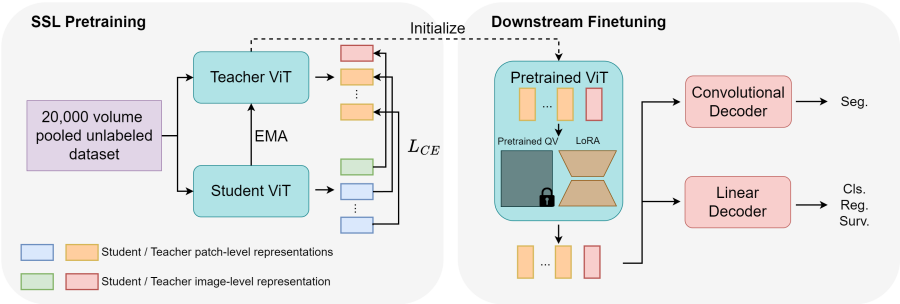

Fig. 1. Diagram summarizing pretraining and downstream finetuning methodology.

2.1 Preprocessing

For SSL pretraining, the only preprocessing steps we use are intensity normalization and foreground cropping. Since we pool CT data with non-quantitative MRI data, we use a simple percentile-based normalization, mapping from 0.05 and 99.95 percentiles to -1 and 1 (clipping values smaller and larger). For fine-tuning, on top of applying intensity normalization and foreground cropping, we also normalize the pixel resolution between scans. The normalized voxel size was determined manually by inspecting each downstream dataset.

2.2 Proposed Method

3DINO We employ 3DINO [17] as our pretraining method of choice. 3DINO is built on DINOv2 [14], which performs SSL using a self-distillation objective utilizing a teacher and student network. From an unlabeled medical image sampled from the dataset, two randomly augmented "crops" are taken (called crop 'A' and 'B'). DINOv2 consists of two primary objectives. The first objective passes crop A to the student, and uses the image-level feature to predict the feature vector output by passing crop B to the teacher (and vice versa passing crop B to the student). The second objective masks patches in crop A, and tasks the student with reconstructing the patch-level features output by passing the unmasked crop A to the teacher. The teacher is parameterized simply as an exponential moving average of the student's weights. More information on this method can be found in the original work [14].

Using 3DINO, we pretrain a vanilla ViT-Large adapted to 3D inputs on the 20,000-volume unlabeled dataset. The input dimensions to the network are $96 \times 96 \times 96$ with a patch size of $16 \times 16 \times 16$. The model has 24 layers and 16 heads, with a feature embedding size of 1024.

LoRA For all downstream tasks, we use LoRA [10] to slightly adapt pretrained weights during fine-tuning. LoRA is a method to fine-tune large models by updating only a small set of extra parameters on top of frozen pretrained weights.

It functions by assuming that downstream task-specific adjustments are intrinsically low-rank. This is implemented by adding a r-rank update to existing pretrained weight matrices. In this work, we apply LoRA to the query and key projection matrices, with rank $r = 16$. More information on this method can be found in the original work [10].

Prediction Tasks For prediction-like tasks that yield a single image-level feature, we append a simple 2-layer linear model to the pretrained ViT-Large model. The linear layer either inputs the `[CLS]` token output of the final ViT layer, or the `[CLS]` token concatenated with the averaged patch tokens. This architecture is consistent with all prediction tasks, with the main difference being the the number of output classes and loss functions. For downstream classification tasks, we use cross-entropy (CE) loss. For regression, we use mean-squared error. For multi-label prediction tasks we use binary CE. Survival prediction employs the negative log-likelihood formulation from [19].

Segmentation Tasks For dense segmentation tasks, we append a convolutional decoder onto the pretrained ViT-Large model similarly to the UNETR [8] architecture. Segmentation tasks use the compound Dice-CE loss due to its robustness for medical image segmentation tasks [13].

Joint Segmentation and Classification Tasks A single downstream task looked to jointly tackle image classification and segmentation. For this task, we used a merged segmentation decoder with a linear classification decoder architecture. The segmentation decoder was trained regularly using Dice-CE loss, with the classification decoder trained using CE loss *only* on randomly sampled patches that contain foreground. Both decoders shared a ViT backbone and LoRA weights. During inference, a sliding window is used to segment foreground regions for a whole image, and a crop centered on the predicted foreground regions is passed to the classification decoder to predict the class of the image.

Resource Efficiency While the pretrained network is relatively large (1.3Gb), performing inference is not costly due to FlashAttention [5] and having lightweight decoders and minimal weight adaptation. Classification weights including LoRA weights are only about 8.1Mb per task, and convolutional decoders including LoRA are about 80Mb per task. With the settings used in this work, inference could be performed on a 8Gb consumer GPU in under 1 minute per case.

3 Experiments

3.1 Dataset and evaluation measures

Details regarding the composition of the pretraining and downstream fine-tuning datasets can be found in the challenge website [5]. A brief summary of the downstream tasks for each modality follows:

CT validation tasks:

1. Abdominal disease classification ('CT Ab. Cls.'): A 24-class multi-label abdominal disease classification task. Performance on this task is determined using mean average precision (mAP).
2. Abdominal lesion segmentation ('CT Ab. Les.'): A 2-class abdominal lesion segmentation task. Performance on this task is determined using the Dice segmentation score (DSC) averaged over the foreground class.
3. Abdominal organ segmentation ('CT Ab. Org.'): A 14-class abdominal organ segmentation task. Performance on this task is determined using DSC averaged over all foreground classes.
4. Lung lesion segmentation ('CT Lung Les.'): A 2-class lung lesion segmentation task. Performance on this task is determined using DSC averaged over the foreground class.

CT testing tasks:

1. Abdominal disease classification: The same 24-class multi-label abdominal disease classification task, except tested on an unseen test set.
2. MSWAL Abdominal lesion segmentation: An 8-class abdominal lesion segmentation task on multiple lesion types.

MRI validation tasks:

1. Brain age prediction ('MRI Age'): A brain age regression task. Performance on this task is determined using mean absolute error (MAE) in years.
2. Neuropsychiatric phenomics classification ('MRI Phen.'): A 4-class neuropsychiatric phenomics prediction task. Performance on this task is determined using balanced accuracy (bal. acc.) and area under the receiver operating characteristic curve (AUC).
3. Autism diagnosis ('ABIDEII'): A 2-class autism diagnosis task. Performance on this task is determined using bal. acc. and AUC.
4. Survival prediction ('UPenn-GBM'): A survival regression task on brain MRI. Performance on this task is determined using the concordance index (C-Index).

[5] https://www.codabench.org/competitions/7150/#/pages-tab

5. Liver tumor segmentation ('ATLAS23'): A 2-class liver tumor segmentation task. Performance on this task is determined using DSC averaged over the foreground class.
6. Heart segmentation and classification ('EMIDEC Seg.' or 'EMIDEC Cls.'): A joint 5-class heart segmentation task and 2-class classification task. Performance on this task is determined using DSC averaged over the foreground classes, bal. acc., and AUC.

MRI testing tasks:

1. Endometriosis classification: A 2-class endometriosis classification task from pelvic MRI with multiple potential input modalities.
2. Abdomen organ segmentation: A 14-class abdominal organ segmentation task on *MRI scans*.

3.2 Implementation details

Environment settings The development environments and requirements for pretraining are presented in Table 1.

Table 1. Development environments and requirements.

CPU	AMD EPYC 7742 64-Core Processor
RAM	300Gb
GPU	Two A100-SXM4-80GB
CUDA version	12.2
Programming language	Python 3.9
Deep learning framework	torch 2.0, torchvision 0.15.0
Specific dependencies	MONAI 1.3.0
Code	https://github.com/AICONSlab/3DINO

Training protocols Pretraining augmentations include a 3D version of RandomResizedCrop [17], random flips and rotation, contrast adjustment, additive noise, histogram shift, Gaussian smoothing and sharpening, and Gibbs noise. Fine-tuning augmentations included random flips and rotation, affine transforms, contrast adjustment, Gaussian smoothing and sharpening, additive noise, and intensity shifts.

For prediction tasks, patches are sampled from full images using a random crop. To minimize the chances of missing important image regions after cropping, image sizes were larger (128^3 or 144^3) and spacing normalization was chosen so that full images were slightly larger than the image size. During inference, a deterministic center crop is taken instead.

For segmentation tasks, patches are sampled from images using random crops that had 50% chance of being sampled from foreground voxels, and 50% chance of

being sampled from background. Image sizes were smaller due to computational complexity of training a convolutional decoder (112^3), and a sliding window was applied to segment a full image during inference.

On downstream validation tasks, best models were selected by the highest performance on the validation sets. For the endometriosis classification testing task, a 3-fold cross-validation scheme was used with the final prediction averaging the outputs of the 3 folds. For CT Ab. Cls., the best performing model on the validation test was chosen. For the two testing segmentation tasks, the final models were used. Additional details on training protocols (pretraining and downstream fine-tuning) are in Tables 2 and 3.

Table 2. Pretraining protocols.

Batch size	256
Patch size	96×96×96
Total iterations	125,000
Optimizer	AdamW
Initial learning rate (lr)	0.001
Lr decay schedule	Linear Warmup Cosine Annealing
Training time	10 days
Loss function	DINOv2 loss [14]
Number of model parameters	300M

Table 3. Training protocols for downstream fine-tuning. If two values, the first refers to segmentation parameters and the second refers to prediction.

Network initialization	3DINO pretraining, model frozen
Batch size	8, 16
Patch size	112×112×112, 128×128×128
Total iterations	Varies per task
Optimizer	AdamW
Initial learning rate (lr)	Varies per task
Lr decay schedule	Linear Warmup Cosine Annealing
Training time	8 hours, Approx. 2 hours
Number of model parameters	20M, 2.5M

4 Results and discussion

4.1 Pre-training Progress

To evaluate the progression of SSL pretraining, we perform linear evaluation using the pretraining weights saved at 5 iteration checkpoints. For prediction tasks, this means training a 2-layer linear network on top of a frozen pretrained model *without* using LoRA. To save time for segmentation, we also only train a 2-layer linear network on top of patch-level features (and do not use LoRA), meaning the resulting segmentation feature maps are 16× downsampled. We evaluate performance by upsampling these feature maps. Since this segmentation decoder is very lightweight, segmentation performance is degraded, though we are mainly looking at relative improvements in performance over pretraining epochs. As a sanity-check baseline comparison, we also add an experiment at iteration 0 (randomly initialized backbone). The results are summarized in Figure 2. Overall, performance seems to increase over pretraining progression.

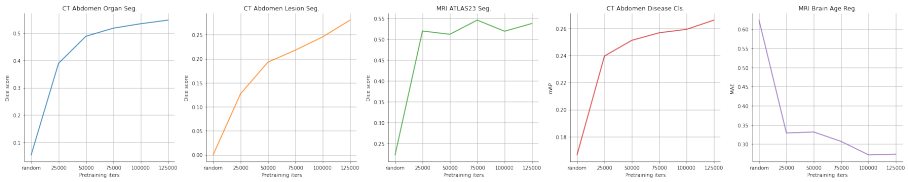

Fig. 2. Linear evaluation results for 5 validation tasks taken at 5 checkpoints every 25000 iterations.

4.2 Use of LoRA

To evaluate the importance of using LoRA to tune pretrained ViT weights, we perform experiments with and without it. These experiments use the same decoder and training recipe, only differing in use of LoRA. Results are summarized in Figure 3. We found LoRA to benefit all tasks except for CT Abdomen Lesion segmentation, where it performed similarly. LoRA benefited MRI Brain Age regression the most, which we expect may be due to the fine-grained nature of the task. Qualitative performance on the CT Abdomen Organ segmentation task are displayed in Figure 4. We found that LoRA appeared to help particularly in detecting smaller organs.

4.3 Quantitative results on validation set

We report results on the validation set of the challenge using the ViT pretrained for 50,000 iterations as the pretrained encoder. We did not use the final pretrained model because it was not done training by the end of the validation phase of the challenge. Performance on segmentation tasks are displayed in Table 4, and prediction tasks in Table 5.

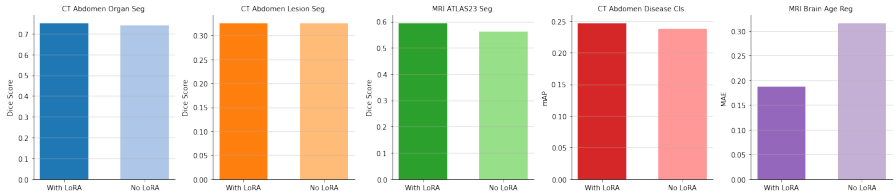

Fig. 3. LoRA results for 5 validation tasks.

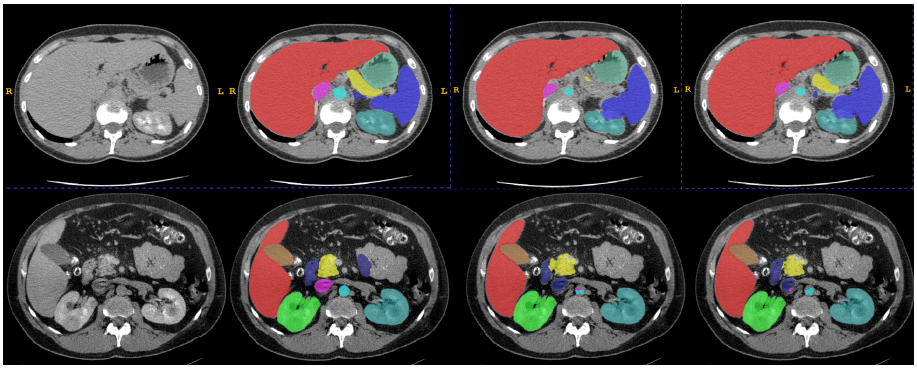

Fig. 4. Performance with and without using LoRA on CT Abdomen Organ segmentation. The two rows are separate cases in the validation set, with columns in order being original image, ground truth, segmentation prediction **without** using LoRA and segmentation prediction **with** LoRA.

Table 4. Quantitative validation segmentation task performance.

Task	CT Ab. Les.	CT Ab. Org.	CT Lung Les.	ATLAS23	EMIDEC Seg.
Metric	DSC	DSC	DSC	DSC	DSC
Value	0.3254	0.7511	0.5798	0.5944	0.4969

Table 5. Quantitative validation prediction task performance.

Task	CT Ab. Cls.	MRI Age	MRI Phen.		ABIDEII		UPenn-GBM	EMIDEC Cls.	
Metric	mAP	MAE	Bal. Acc.	AUC	Bal. Acc.	AUC	C-Index	Bal. Acc.	AUC
Value	0.2469	3.262	0.4104	0.6423	0.6439	0.6439	0.5967	0.8439	0.8972

4.4 Results on final testing set

Final testing results will be added after their release.

4.5 Limitation and future work

One limitation of our work is that it appears to struggle with segmentation tasks that contain smaller foreground regions (e.g. lesion segmentation). This may be because ViTs inherently operate on the downsampled patch-level. While LoRA did appear to help with this, additional work incorporating adapters, pretraining vision-specific ViTs like Sliding Window Transformer [12], or distilling ViT weights to convolutional encoders could further improve performance.

Another limitation of this work is that it used a relatively barebones preprocessing and training recipe, and did not use any postprocessing. Additional work on this, perhaps incorporating the nnUNet [11] framework may improve our segmentation performance.

5 Conclusion

In this work, we apply the 3DINO pretraining method to generate representations from the pooled 20,000-scan FLARE Challenge Task 4 unlabeled dataset. To better adapt the pretrained model to downstream tasks without overfitting and with minimal additional trained weights, we utilize LoRA. We show that pretraining and LoRA benefits performance in a large variety of downstream tasks in medical imaging including segmentation, classification, survival prediction, and regression. Overall, the creation of SSL methods that are capable of generating salient features for 3D medical images could greatly reduce the amount of labeled data needed for diverse downstream medical imaging tasks.

Acknowledgements The authors of this paper declare that the proposed solution is fully automatic without any manual intervention. We thank all data owners and contributors for making the data publicly available and CodaLab [18] for hosting the challenge platform. This work was supported by funding from the Natural Sciences and Engineering Research Council (NSERC) Discovery grant (RGPIN-2021-03728), Canada Foundation for Innovation (40206), and the Ontario Research Fund. T.X. is funded by the NSERC PGS-D award and Google PhD Fellowship. M.G. is funded by the Canada Research Chairs program award (CRC-2021-00374). A.L.M. is supported by the Tory Family Chair in Oncology.

Disclosure of Interests

The authors declare no competing interests.

References

1. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021) 1
2. Chen, R.J., Ding, T., Lu, M.Y., Williamson, D.F., Jaume, G., Song, A.H., Chen, B., Zhang, A., Shao, D., Shaban, M., et al.: Towards a general-purpose foundation model for computational pathology. Nature medicine **30**(3), 850–862 (2024) 2
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 1
4. Chen, Z., Agarwal, D., Aggarwal, K., Safta, W., Balan, M.M., Brown, K.: Masked image modeling advances 3d medical image analysis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1970–1980 (2023) 2
5. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in neural information processing systems **35**, 16344–16359 (2022) 4
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 2
7. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020) 1
8. Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: Unetr: Transformers for 3d medical image segmentation. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 574–584 (2022) 4
9. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022) 1
10. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022) 3, 4
11. 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) 10
12. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021) 10
13. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**, 102035 (2021) 4
14. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 2, 3, 7

15. Taleb, A., Loetzsch, W., Danz, N., Severin, J., Gaertner, T., Bergner, B., Lippert, C.: 3d self-supervised methods for medical imaging. Advances in neural information processing systems **33**, 18158–18172 (2020) 2
16. Vorontsov, E., Bozkurt, A., Casson, A., Shaikovski, G., Zelechowski, M., Liu, S., Severson, K., Zimmermann, E., Hall, J., Tenenholtz, N., et al.: Virchow: A million-slide digital pathology foundation model. arXiv preprint arXiv:2309.07778 (2023) 2
17. Xu, T., Hosseini, S., Anderson, C., Rinaldi, A., Krishnan, R.G., Martel, A.L., Goubran, M.: A generalizable 3d framework and model for self-supervised learning in medical imaging. arXiv preprint arXiv:2501.11755 (2025) 2, 3, 6
18. Xu, Z., Escalera, S., Pavão, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022) 10
19. Zadeh, S.G., Schmid, M.: Bias in cross-entropy-based training of deep survival networks. IEEE transactions on pattern analysis and machine intelligence **43**(9), 3126–3137 (2020) 4
20. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021) 1

Table 6. Checklist Table. Please fill out this checklist table in the answer column.

Requirements	Answer
A meaningful title	Yes/No
The number of authors ($\leq$6)	3
Author affiliations and ORCID	Yes/No
Corresponding author email is presented	Yes/No
Validation scores are presented in the abstract	Yes/No
Introduction includes at least three parts: background, related work, and motivation	Yes/No
A pipeline/network figure is provided	Fig. 1
Pre-processing	Pg. 3
Strategies to use the partial label	Pg. 4
Strategies to use the unlabeled images.	Pg. 3
Strategies to improve model inference	Pg. 4
Post-processing	N/A
The dataset and evaluation metric section are presented	Pg. 5
Environment setting table is provided	Tab. 1
Training protocol table is provided	Tab. 2, 3
Ablation study	Pg. 8
Efficiency evaluation results are provided	Pg. 4
Visualized segmentation example is provided	Fig. 4
Limitation and future work are presented	Yes/No
Reference format is consistent.	Yes/No