IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 30 January 2024, accepted 18 April 2024, date of publication 29 April 2024, date of current version 8 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3395058

RESEARCH ARTICLE

QT-UNet: A Self-Supervised Self-Querying All-Transformer U-Net for 3D Segmentation

ANDREAS HAMMER HÅVERSEN, DURGA PRASAD BAVIRISETTI, GABRIEL HANSSEN KISS, AND FRANK LINDSETH
Department of Computer Science, Norwegian University of Science and Technology, Trondheim, 7034 Trøndelag, Norway
Corresponding author: Durga Prasad Bavirisetti (durga.bavirisetti@ntnu.no)

ABSTRACT With reliable performance, and linear time complexity, Vision Transformers like the Swin Transformer are gaining popularity in the field of Medical Image Computing (MIC). Examples of effective volumetric segmentation models for brain tumours include VT-UNet, which combines conventional UNets with Swin Transformers using a unique encoder-decoder Cross-Attention (CA) paradigm. Self-Supervised Learning (SSL) has also experienced an increase in adoption in computer vision domains such as MIC, in situations where labelled training data is scarce. The Querying Transformer UNet (QT-UNet) model we introduce in this paper brings these advancements together. It is an all-Swin Transformer UNet with an encoder-decoder CA mechanism strengthened by SSL. For the purpose of evaluating the potential of QT-UNet as a generic volumetric segmentation model, it is subjected to extensive testing on several MIC datasets. Our best model achieves a Dice score of 88.61 on average and a Hausdorff Distance of 4.85mm making it competitive with State of the Art in Brain Tumour Segmentation (BraTS) 2021, using 40% fewer FLOPs than the baseline VT-UNet. We found poor results with Beyond The Cranial Vault (BTCV) and Medical Segmentation Decathlon (MSD), but validate the effectiveness of our new CA mechanism and find that the SSL pipeline is most effective when pre-trained with our CT-SSL dataset. The code be can found at https://github.com/AndreasHaaversen/QT-UNet.

INDEX TERMS Deep learning, encoder-decoder cross-attention, UNet, medical image segmentation, self-supervised learning, Swin Transformer, vision transformer.

I. INTRODUCTION

Transformers, introduced by Vaswani et al. [1], revolutionised the Natural Language Processing (NLP) field by introducing a model that can effectively model long-range dependencies while maintaining a manageable computational cost. Transformers are now the dominant model in that field, with notable examples being BERT [2] and GPT-3 [3]. The computational efficiency of the Transformer has enabled NLP models of unprecedented size, with the largest variant of GPT-3 having approximately 175 billion trainable parameters.

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

The attention mechanisms that power Transformers have also inspired the adoption of similar mechanisms in models for Computer Vision (CV), with some models incorporating self-attention mechanisms instead of convolution or using a Transformer in conjunction with a convolutional backbone.

In 2020, Dosovitskiy et al. [4]. introduced the Vision Transformer (ViT), a near-end-to-end image classification model. The results demonstrated that Transformers can handle vision tasks without extensive backbones or Convolutional Neural Network (CNN)'s inductive bias. These models outperformed State-of-the-Art CV models in image classification while using fewer computing resources and parameters.

Nevertheless, traditional Transformer models suffer a quadratic rise in memory and time complexity based on input length. This makes Transformers difficult to employ on high-resolution images and volumes, such as those required for Medical Image Computing (MIC). Several methods [5], [6], [7] have been suggested to reduce the time complexity of traditional Transformers.

The Swin Transformer was proposed by Liu et al. [8] which achieved linear-time complexity by utilising windowed self-attention and a shifting mechanism. This has allowed its use in tasks involving both high-resolution images and volumetric data, such as CT and MRI scans. Due to its effectiveness, the Swin Transformer has been used in a variety of dense prediction applications, including segmentation and depth estimation. Two such examples are Swin-UNet [9] and VT-UNet [10].

Swin-UNet [9] uses Swin Transformer blocks in a UNet architecture. It is mainly used with CT scans and outperformed convolution-based UNets in these MIC tasks. However, this method is limited to 2D slices, and there is a significant loss of 3D context during the segmentation process.

VT-UNet [10] is a method for volumetric segmentation of MRI images inspired by Swin-UNet, which employs video Swin Transformer blocks inside a UNet architecture. Moreover, a new Cross-Attention (CA) mechanism is introduced between the encoder and the decoder, with the encoder providing the decoder with keys and values at each stage. This method performs well on Brain Tumour Segmentation (BraTS) 2021 data while being much smaller in terms of parameters and using fewer computing resources than comparable models like UNETR [11], nnUNet [12], and nnFormer [13]. However, the keys and values that the encoder sends to the decoder are fixed, irrespective of the needs of the decoder. The decoder block also uses a parallel branch scheme that is computationally intensive, but contributes negligibly to model performance.

UNETR [11] is another 3D UNet architecture for segmenting CT images that combines a ViT-based encoder with a traditional CNN decoder. It is now one of the top-performing methods on the Beyond The Cranial Vault (BTCV) dataset. However, the encoder uses the same feature resolution throughout, indicating that the method lacks a correct feature hierarchy.

When dealing with a machine learning problem, one of the key challenges is collecting enough data to train the model without overfitting and poor generalisation. This is especially difficult in the field of MIC, since labelling the data is a time-consuming process that requires the expertise of qualified professionals. By automatically generating pseudo-labels for easily accessible unlabelled data, Self-Supervised Learning (SSL) enables ML practitioners to get more out of their data without labelling. To great success, this has been used in CV and, more specifically, MIC. Swin-UNETR [14] is a remarkable work that integrates SSL with a UNet using a 3D Swin Transformer as its encoder.

Swin-UNETR [14] builds upon UNETR and combines a Swin Transformer-based encoder with a CNN decoder. Notably, for this encoder, authors provide a revolutionary SSL approach that incorporates contrastive learning, masked volume in-painting, and 3D rotation prediction SSL heads. SSL training is carried out utilising augmented sub-volumes of CT scans, which are enhanced by rotating the samples in the z-axis and applying random sub-volume masking. In addition, the authors collected a large dataset for use with the SSL system.

This study builds upon recent works on Vision Transformers, focusing on MIC. As VT-UNet and Swin-UNETR are strong models for a number of MIC datasets, we draw inspiration from them and attempt to combine and improve upon their respective approaches. Thus, we present QT-UNet, which accomplishes this goal through an all-Swin Transformer UNet specially designed with Encoder-Decoder CA and SSL.

A. RESEARCH PROBLEMS

In this study, we examine the impact of implementing a generic cross-modality model on 3D data from a wide range of sources, using CA as realised in VT-UNet and SSL as proposed in Swin-UNETR. In order to achieve better model performance, we are also looking for ways to enhance their original methods. The overall purpose of our study is to:

> Test the performance of a cross-domain all-Transformer UNet segmentation model trained using the Swin Transformer, self-supervised pre-training, and Encoder-Decoder CA on datasets in the MIC field.

To achieve this goal, we investigate the following research questions (RQs):

- **RQ1:** What is the effect of using self-supervised pretraining of the encoder in an all-Transformer UNet on the performance of the overall network in segmentation tasks?
- **RQ2:** What is the effect of using encoder-decoder CA on the overall performance of an all-Transformer UNet?

II. CONTRIBUTIONS

We introduce the Querying Transformer UNet (QT-UNet), leveraging SSL and and CA to create a Swin-based all-Transformer U-Net for semantic segmentation of 3D data. In order to evaluate its performance, we subject it to a series of experiments using MIC datasets, and comparing it with the latest state-of-the-art models for each dataset.

We introduce a novel CA mechanism inspired by VT-UNet [10], coupled with a new decoder block design that allows the decoder blocks in the model to query the output of the same-stage encoder for information at each stage of the decoding process. We also employ SSL for the encoder, based on the procedure developed for Swin-UNETR [14]. We collect a large dataset consisting of 3,597 CT scans, dubbed CT-SSL, to pre-train the encoder for CT-based tasks.

When trained on the BraTS2021, we see a significant improvement, and when trained with pre-learned weights on this dataset, we observe even better performance in terms of Hausdorff Distance. Both models are competitive and achieve a 40% reduction in FLOPs when compared to the baseline VT-UNet. Experiments with BTCV and MSD yield weaker results, though they validate the effectiveness of the new CA technique, and our SSL pipeline when pretraining with CT-SSL.

The structure of the remaining paper is as follows: The Methods section presents an in-depth explanation of QT-UNet. Experiments and Results are described in the succeeding section. Finally, concluding remarks and future research directions are discussed in the last section.

III. METHODS

The proposed QT-UNet is shown in Figure 1. The overall model architecture and training procedure is inspired by VT-UNet [10] and Swin-UNETR [14]. It takes a 3D volume $D \times H \times W \times C$ and produces a volume of size $D \times H \times W \times K$, where K is the number of target classes. A detailed explanation of each component of the proposed method is presented as follows:

A. QT-UNET ENCODER

The QT-UNet encoder consists of a 3D patch partitioning layer and a linear embedding layer, followed by successive QT encoder blocks and patch merging layers. Each stage in the encoder consists of two QT encoder blocks, followed by a patch merging layer.

1) PATCH PARTITIONING AND LINEAR EMBEDDING LAYER

The first layer of QT-UNet, like other ViTs, takes the input volume and generates a sequence of tokens by splitting the input into non-overlapping patches using a convolutional layer. Each kernel in the layer has size $M \times M \times M$, producing a sequence of $\lfloor \frac{D}{M} \rfloor \times \lfloor \frac{H}{M} \rfloor \times \lfloor \frac{W}{M} \rfloor$ tokens describing the volume. These tokens are then flattened by a linear embedding of each token with dimensionality $M \times M \times M$ to a C dimensional vector.

2) QT ENCODER BLOCK

The QT encoder block draws upon the design of the Video Swin encoder block [15] and VT encoder block [10]. Each block has two sub-blocks utilising a 3D windowed Multi-Head Self-Attention (W-MHSA) module followed by a two-layer MLP with GELU activation. Layer normalisation is performed before and after the W-MHSA module, with skip connections across both the W-MHSA and MLP modules. For the second sub-block, a two voxel shift is applied in each direction before windowing to introduce cross-window connections between the blocks. This shifted operation is known as Shifted Window Multi-Head Self-Attention (SW-MHSA). Finally, in each self-attention head, a relative bias of $B \in \mathbf{R}^{M^2 \times M^2 \times M^2}$ is applied.

(1) describes self-attention as applied in each window, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbf{R}^{M^3 \times d}$ are the query, key, and value matrices, and d is the dimension of the key and value features.

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\intercal}}{\sqrt{d_q}} + \boldsymbol{B}\right)\boldsymbol{V} \quad (1)$$

Since the relative position along each axis lies in the range of $[-M+1, M-1]$, we parameterise a smaller bias matrix $\hat{B} \in \mathbf{R}^{(2M-1)\times(2M-1)\times(2M-1)}$, taking values for B from $\hat{B}$, as in [8].

The windowing operation can be understood as injecting an inductive bias for locality into the model. The shifting operation allows successive applications of the blocks to receive information across windows, while the position bias informs the relative positioning of those windows.

3) PATCH MERGING

Strong feature hierarchies are essential to most segmentation models [8], [9], [10], [14], [16], [17]. To achieve this in the QT encoder, adjacent $2 \times 2 \times 2$ tokens are concatenated along their feature dimension after each stage, giving dimensions $D/2 \times H/2 \times W/2 \times 8C$. A linear layer is used to shrink the concatenated features to one-fourth of their expanded dimension ($8C \rightarrow 2C$), resulting in a final volume of $D/2 \times H/2 \times W/2 \times 2C$.

Our application of patch merging differs slightly from the equivalent mechanism in the Video Swin Transformer [15] and VT-UNet [10], since QT-UNet merges adjacent tokens in all three spatial axes, not just height and width.

B. BOTTLENECK

The bottleneck layer is the deepest layer of the model and consists of a single QT encoder block, followed by a patch expansion layer.

C. QT-UNET DECODER

The QT-UNet decoder consists of successive pairs of patch expansion layers and QT decoder blocks and ends with a classifier.

1) PATCH EXPANSION

The patch expansion layers do the opposite of the patch merging layers. They increase the spatial resolution of tokens while decreasing the size of their features.

This is a two-stage process: First, a linear layer expands the feature dimensions fourfold ($2C \rightarrow 8C$). Then, $2 \times 2 \times 2$ tokens with feature dimension C are extracted from the expanded token, producing a sequence of tokens corresponding to a volume of size $D \times H \times W \times C$.

2) QT DECODER BLOCK

Iterating upon the VT decoder block from VT-UNet [10], the QT decoder block introduces two significant changes.

First, instead of using keys and values generated in the encoder, the QT decoder block generates them from the

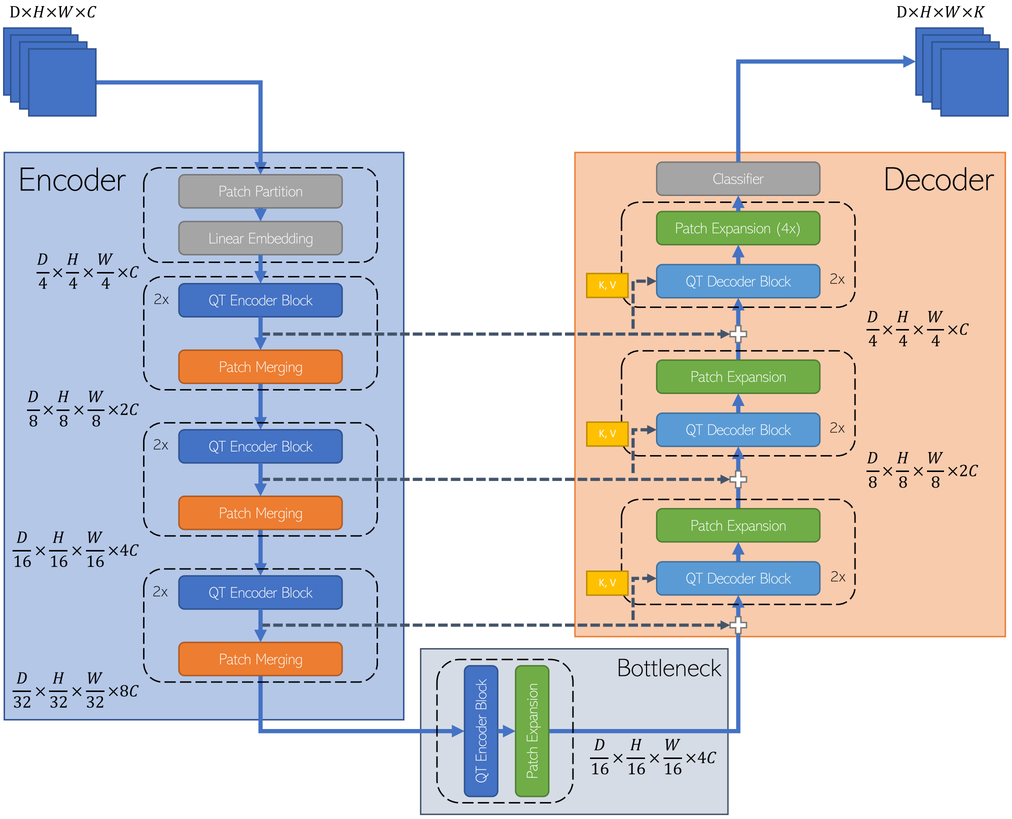


FIGURE 1. The proposed model, QT-UNet.

output of the same-stage encoder block. This allows each decoder block to query the spatially dense output of the same stage encoder block more flexibly while saving compute at the cost of more parameters.

Second, the QT decoder block replaces the Fusion Module from VT-UNet with a more conventional decoder design reminiscent of the original Transformer decoders due to Vaswani et al. [1]. First, standard W-MSA is applied with keys, queries, and values derived from the block input. Then, windowed CA is applied, generating keys and values from the output of the same-stage encoder and generating queries from the previous self-attention block.

The QT decoder block and its interaction with the encoder block is illustrated in Figure 2. Similar to the encoder blocks, there are several skip connections across the modules in each sub-block, with each sub-block being topped with a two-layer MLP with GELU activation. Windows are shifted 2 voxels in each axis for each pair of sub-blocks to produce shifted window self-attention. A relative spatial bias is also applied in the same manner as in the encoder.

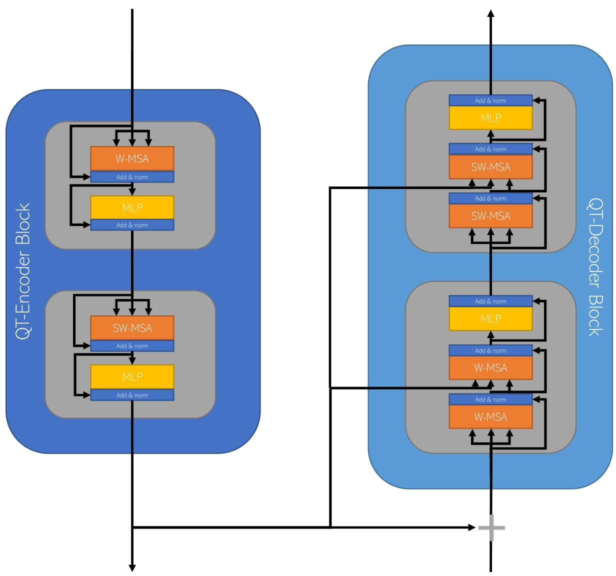


FIGURE 2. Overview of the interaction between encoder and decoder stages.

D. CLASSIFIER

Following a final patch expansion layer in the decoder, the model is topped with a convolutional classification head, mapping the C dimensional features to K segmentation classes.

E. VARIANTS

Three variants of QT-UNet are proposed by adjusting C, the number of embedding dimensions in the patch embedding layer. Using the same naming convention as VT-UNet [10], the three variants are as follows:

1) Tiny: **QT-UNet-T**, $C = 48$
2) Small: **QT-UNet-S**, $C = 72$
3) Base: **QT-UNet-B**, $C = 96$

All models employ three stages of encoding and decoding, plus the bottleneck.

F. COMMON PARAMETERS

All model variants use a patch embedding size of $M = 4$. A window size of $7 \times 7 \times 7$ is used for window partitioning. Additionally, the number of heads in each module follows the pattern given in (2), increasing as the input descends into the encoder, reaching maximum of 24 in the bottleneck, and decreasing as it ascends through the decoder.

The -S and -T QT-UNet variants have their weights randomly initialised according to [18], though the encoder modules in QT-UNet-B are preloaded with Swin Transformer weights pretrained on ImageNet following [10].

The model is trained by minimising Dice Loss.

$$3 \rightarrow 6 \rightarrow 12 \rightarrow 24 \rightarrow 12 \rightarrow 6 \rightarrow 3 \tag{2}$$

G. INFERENCE WITH QT-UNET

QT-UNet uses sliding window inference in constant mode with an overlap of 0.5 to process full size input volumes during validation and testing.

H. SSL IN QT-UNET

SSL is employed on the encoder before fine-tuning using an approach built upon the one used by [14] for Swin-UNETR. Similar to it, QT-UNet is pre-trained using an augmented multi-view multi-head approach. First, a sub-volume $x \in \mathbf{R}^{d\times h\times w\times C}$[1] is extracted from the larger input volume $X \in \mathbf{R}^{D\times H\times W\times C}$. Two augmented views of the data are generated from this sub-volume x, with two independent applications of an augmentation pipeline consisting of random sub-volume masking and random 90° rotation along the z-axis. These augmented views are then passed to the encoder, whose output is passed to each of the heads described in the following three subsections, optimising the encoder by the joint loss of the heads.

1) RECONSTRUCTION HEAD

Consisting of a single transposed convolution layer, this head takes the view representation as input and attempts to reconstruct the un-augmented sub-volume x. Loss is calculated using L1 loss between the reconstruction $\hat{x}$ and x.

2) IMAGE ROTATION HEAD

Consisting of a standard one layer MLP with Batch Norm and a ReLU activation function, this head predicts how much the augmented volume was rotated: Either 0°, 90°, 180°, or 270°. Loss is calculated using the soft-maxed cross-entropy between the true rotation k and the prediction $\hat{k}$.

[1] Where d, h, and w are the spatial dimensions of the volume.

3) BOOTSTRAP YOUR OWN LATENT (BYOL) HEAD

Swin-UNETR uses a SimCLR-based approach that requires large batch sizes to be effective [19]. Noting that the batch sizes are severely limited by memory usage, we instead use BYOL [20] in this head due to its superior performance with smaller batch sizes.

The head is based on the BYOL implementation provided by PyTorch Lightning Bolts [21], with modifications to fit the augmentation scheme. Loss is calculated using the cosine similarity between the outputs of the online and target branches.

4) MODES OF OPERATION FOR QT-UNET SSL

The SSL setup is used in two different modes, depending on how much data is available for pretraining in the specific task and modality. In tasks with sufficient data, we perform pretraining with a large out-of-task dataset, referring to it as ‘‘out-of-task pretraining’’. Otherwise, pretraining is performed using the task data directly, referring to it as ‘‘in-task pretraining’’. For more details and in-depth explanation of the proposed QT-UNet method, please refer to our work [22].

IV. EXPERIMENTS AND RESULTS

This section presents the software and hardware that were used to develop the methodology and run the experiments. It also details the experiments that were run and the various datasets that were utilised for the study. Finally, the experimental results are thoroughly investigated.

A. SOFTWARE AND HARDWARE

Our experimental setup was configured using Anaconda [23] with Python 3.9.11, PyTorch 1.11.0 [24], PyTorch Lightning 1.6.0 [21], PyTorch Lightning Bolts 0.5.0 [25], and MONAI 0.8.1 [26]. IDUN, an HPC cluster at NTNU, was used for training [27]. A NVIDIA A100 40GB GPU was used for training runs, with two A100 80GB cards being used for SSL runs.

B. QUANTITATIVE METRICS

Two standard segmentation metrics Dice Score [28] and Hausdorff Distance (HD) [29] are used for quantitative analysis.

C. DATASETS

A detailed explanation of various datasets used and SSL pretraining experiments are presented in the following subsections.

1) CT-SSL DATASET

We compiled ‘‘CT-SSL’’, a large CT dataset of abdomen, pelvis, and chest scans, using publicly available datasets from The Cancer Imaging Archive (TCIA) [30]. Using the TCIA API, we downloaded 3597 CT scans and converted them to Nifti format, with 100 scans serving as a validation set during training. Table 1 summarises the datasets contained in CT-SSL.

TABLE 1. Overview of datasets in CT-SSL.

Dataset	Region	#of scans
CT Lymph Nodes [31]	Abdomen/Lungs	175
CT Colonography [32]	Abdomen/Pelvis	1706
COVID-19-AR [33]	Lungs	149
MIDRC-RICORD-1A [34]	Lungs	121
MIDRC-RICORD-1B [35]	Lungs	90
Pelvic Reference Data [36]	Pelvis	116
Stage II Colorectal CT [37]	Abdomen/Pelvis	230
LiDC [38]	Chest	1010

TABLE 2. SSL training parameters.

Parameter	Value
Learning Rate	0.4×10^{-4}
Weight decay	1.5×10^{-6}
Optimiser	Adam
Learning rate scheduler	Linear Warm-up Cosine Annealing
Mini-batch Size	Varies with experiment, see Table 4
Epochs	Varies with experiment, see Table 3
Warm-up epochs	10

TABLE 3. SSL epochs.

Dataset	Num epochs
CT-SSL	150
BraTS2021	350
MSD Task 2/4/5	350

2) PREPARATORY SSL

QT-UNet was pretrained using our SSL setup with CT-SSL, BraTS2021 and tasks 2, 4, and 5 from MSD, with hyper-parameters listed in Table 2.

The number of epochs is determined by whether the dataset is utilised for in-task or out-of-task pretraining. Because the out-of-task dataset CT-SSL is quite large, we chose fewer epochs due to its size and time constraints whereas we used a larger number of epochs with smaller in-task datasets to extract as much learning as possible from the data. Table 3 shows the epochs chosen.

In order for BYOL to perform well, GPU batch sizes with gradient accumulation were tuned to fit as many samples as possible. Table 4 displays the effective batch size for each dataset.

a: CT-SSL

CT-SSL is used to pretrain all variants of QT-UNet for downstream CT-based tasks. Each scan is interpolated to an isotropic voxel spacing of $[1.0 \times 1.0 \times 1.0]mm$, before cropping out zero-valued foreground and normalising the values. Random $96 \times 96 \times 96$ subvolumes are then passed to the SSL pipeline.

b: BRATS 2021

Pretraining for BraTS is performed in-task for all QT-UNet variants due to the scarcity of relevant data. The data augmentation process is the same as in subexperiment 1.

c: MSD

In-task pre-training for MSD MRI tasks with all variants of QT-UNet was limited by the availability of relevant out-of-task data for Tasks 2, 4, and 5. The augmentation pipeline is the same as for the regular training runs, but includes only spacing, foreground cropping, clipping, normalisation, and sample extraction augmentations.

D. EXPERIMENTS

Our main experiment has three subexperiments, as detailed below. All variants of QT-UNet and VT-UNet employ the same hyperparameters listed in Table 5. QT-UNet variants are also trained with pretrained weights relevant for the given experiment. The number of FLOPs needed for a forward pass and model size (parameters) are reported in all experiments. FLOPs for VT-UNet and QT-UNet are recorded using fvcore [39] with a forward pass in training mode.

a: SUBEXPERIMENT 1: BRATS 2021

The dataset contains 1251 MRI scans of dimensions $240 \times 240 \times 150$ from several institutions using different equipment and protocols. Following [10], we split the dataset into 834, 208, and 209 scans for training, validation, and testing respectively.

Each scan is interpolated to $[1.0 \times 1.0 \times 1.0]mm$ isotropic voxels. The zero-valued foreground is cropped, and non-zero intensities are normalised channel-wise. For training, randomly selected sub-volumes of size $128 \times 128 \times 128$ voxels are employed.

Following standard preprocessing for the BraTS dataset, we use the original labels to produce three classes: Enhancing Tumour (ET), Tumour Core (TC) (Non-Enhancing Tumour + Necrotic Tumour + ET), and Whole Tumour (WT) (Peritumoral edema + TC).

Pretrained variants of QT-UNet utilise weights pretrained directly on BraTS 2021.

We report Dice score and 95th percentile Hausdorff Distance for each class as average values on our local test split and against the BraTS Continuous Evaluation server.

b: SUBEXPERIMENT 2: BTCV

This dataset contains 50 CT scans, 40 of which are labelled with 13 organ segmentation targets. Each CT scan has 85 to 198 slices with 512×512 pixels. Of the 40 labelled scans, 35 are used for training, with the remainder being used for validation and testing.

Due to commonalities in our training scenario, our pre-processing pipeline follows the Swin-UNETR [14]. Before clipping and normalising intensities between -175 and 250, we interpolate each image to $[1.5 \times 1.5 \times 2.0]mm$ voxels. Zero-valued foreground is cropped, and the labels are one-hot encoded. For training, we extract $96 \times 96 \times 96$ voxel subvolumes and perform stochastic augmentations: random flips in each dimension with probability 0.1, random 90 degree rotation with probability 0.1, and finally random intensity shift with offset 0.1 and probability 0.5.

Pretrained variants of QT-UNet utilise weights pretrained on CT-SSL.

TABLE 4. SSL batch sizes.

Dataset	Batch size	Gradient accumulation	Num. GPUs	Effective batch size
CT-SSL	32	2	2	128
BraTS2021	8	4	2	64
MSD Task 2/4/5	32	2	2	128

TABLE 5. Training parameters.

Parameter	Value
Learning Rate	0.4×10^{-4}
Weight decay	0
Drop path rate	0.2
Optimiser	Adam
Learning rate scheduler	Cosine Annealing
Mini-batch Size	1
Epochs	350

TABLE 6. Mapping between MSD tasks and datasets used for pretraining.

Task	Modality	Dataset used for pretraining
1 Brain Tumour	MRI	BraTS2021
2 Heart	MRI	MSD Task 2
3 Liver	CT	CT-SSL
4 Hippocampus	MRI	MSD Task 4
5 Prostate	MRI	MSD Task 5
6 Lung	CT	CT-SSL
7 Pancreas	CT	CT-SSL
8 Hepatic Vessel	CT	CT-SSL
9 Spleen	CT	CT-SSL
10 Colon	CT	CT-SSL

We present results against our local test split, with leaderboard results provided in a separate table for context. We were unable to make our own submission, as the BTCV leaderboard is currently not processing new submissions.

c: SUBEXPERIMENT 3: MSD

Similar to BCTV, the pre-processing pipelines used for Swin-UNETR [14] are also applied here due to similarities in the training setup. We use the default data splits for each task provided by MONAI [26].

Pretrained variants of QT-UNet are initialised with weights relevant for each task, as given in Table 6.

We report Dice score on the validation set for each task whilst ignoring the background label. As the submissions for the Medical Segmentation Decathlon are closed, we were unable to make our own submission to the leaderboard. We nevertheless include the top leaderboard results in this paper for context.

E. RESULTS

The results of the subexperiments along with the ablation study are presented below.

1) SUBEXPERIMENT 1: BRATS2021

Results of BraTS2021 are reported in Table 8, with qualitative results in Figure 3. It can be noted that QT-UNet uses the fewest FLOPs whilst attaining comparable Dice results. QT-UNet-B also attains the 2nd best average Dice score, with pretrained variants of QT-UNet attaining a lower Hausdorff Distance than those trained from scratch. As show in Figure 3, along the edges, QT-UNet also qualitatively provides a significantly finer segmentation mask than VT-UNet, with pre-trained QT-UNet providing an even tighter mask.

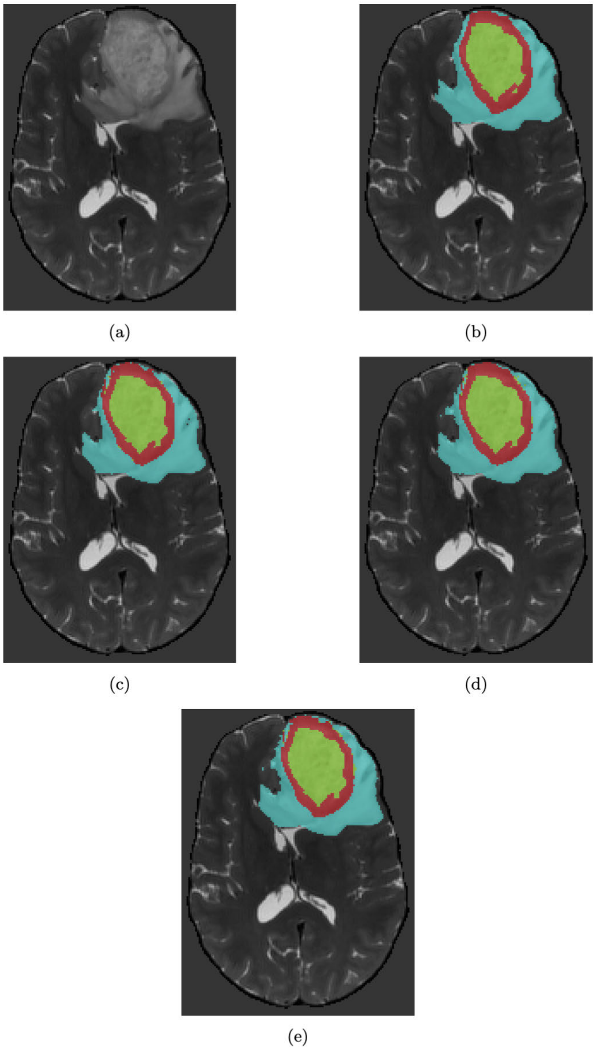


FIGURE 3. Qualitative analysis of various models on one sample from BraTS2021. (a) Raw image. (b) Ground truth. (c) VT-UNet-B. (d) QT-UNet-B/ scratch (e) QT-UNet-B. Green = Tumour Core, Red = Enhancing Tumour, Turquoise = Whole Tumour.

2) SUBEXPERIMENT 2: BTCV

The results of BTCV per organ in Table 9 and the average in Table 11 show that QT-UNet /scratch performs significantly better than VT-UNet, especially for the smaller organs such as the oesophagus, the aorta, the inferior vena cava, portal and spleinc veins, the pancreas, and adrenal glands with 23.5 Dice point margins on average. QT-UNet /scratch is also significantly better than VT-UNet at distinguishing the

TABLE 7. BraTS2021 results on our data split.

Model	Params ↓	FLOPs ↓	Metrics avg.		Dice score ↑				Hausdorff Distance ↓			
			Dice ↑	HD ↓	ET	TC	WT	Avg.	ET	TC	WT	Avg.
VT-UNet-T	**5.4** M	52.0 G	88.68	**4.69**	85.71	**88.40**	91.94	88.68	**4.12**	4.73	5.12	**4.69**
VT-UNet-S	11.8 M	100.8 G	88.82	5.10	**86.58**	88.01	91.85	88.82	4.16	**4.68**	5.35	5.10
VT-UNet-B	20.8 M	165 G	88.64	4.71	86.11	87.88	91.89	88.63	4.22	5.13	**4.53**	4.71
QT-UNet-T /scratch	6.4 M	**32.5** G	88.15	5.79	85.38	87.00	92.06	88.15	4.32	6.24	6.48	5.79
QT-UNet-S /scratch	14.5 M	61.3 G	88.56	5.39	85.90	87.60	**92.20**	88.56	4.18	5.17	5.95	5.39
QT-UNet-B /scratch	25.5 M	98.5 G	**88.69**	4.92	86.27	87.61	92.19	**88.69**	4.23	5.14	5.21	4.92
QT-UNet-T	6.4 M	**32.5** G	88.03	5.31	84.58	87.42	92.09	88.03	4.40	5.74	5.54	5.31
QT-UNet-S	14.5 M	61.3 G	88.50	5.18	85.66	87.70	92.15	88.50	4.43	5.63	5.24	5.18
QT-UNet-B	25.5 M	98.5 G	88.61	4.85	85.61	87.78	92.10	88.61	4.23	4.99	5.23	4.85

ET = Enhancing Tumour, TC = Tumour Core, WT = Whole Tumour.

TABLE 8. BraTS2021 validation results as reported by online evaluation server (BraTS continuous evaluation).

Model	Metrics avg.		Dice score ↑				Hausdorff Distance ↓			
	Dice ↑	HD ↓	ET	TC	WT	Avg.	ET	TC	WT	Avg.
Swin-UNETR	88.97	**5.21**	85.80	**92.60**	88.50	88.97	**6.02**	**5.83**	3.77	**5.21**
Extended nnUNet[a] [40]	88.36	10.61	84.51	87.81	**92.75**	88.36	20.73	7.623	**3.47**	10.61
NVAUTO [41]	**89.11**	6.16	**86.00**	88.68	92.65	**89.11**	9.05	5.84	3.60	6.16
CNN ensemble [42]	87.81	9.58	84.10	87.33	92.00	87.81	16.02	8.91	3.81	9.58
QT-UNet-T /scratch	84.24	11.64	79.30	82.62	90.81	84.24	19.07	10.68	5.19	11.64
QT-UNet-S /scratch	83.61	15.27	77.82	82.00	91.01	83.61	25.79	14.59	5.42	15.27
QT-UNet-B /scratch	84.51	12.26	79.59	82.63	91.32	84.51	19.84	12.36	4.59	12.26
QT-UNet-T	84.19	12.64	78.43	83.27	90.86	84.19	22.29	11.11	4.52	12.64
QT-UNet-S	84.25	12.77	79.25	82.23	91.26	84.25	18.21	15.58	4.51	12.77
QT-UNet-B	84.81	11.53	79.99	83.20	91.24	84.81	17.19	12.95	4.44	11.53

ET = Enhancing Tumour, TC = Tumour Core, WT = Whole Tumour.
[a]BraTS2021 challenge winner

left kidney from the right, with a 20.39 Dice point margin between the two models on average compared to 3 points difference for the right kidney.

Pretraining gives QT-UNet-T a significant 13 point performance boost, and a 3 point boost for QT-UNet-B. QT-UNet-S sees little to no change. Qualitative results in Figure 4 show that VT-UNet struggles significantly, misclassifying the liver and spleen as parts of the stomach. Neither model is able to correctly segment the fluid-filled stomach, whereas QT-UNet is able to correctly segment several organs. The pretrained variant of QT-UNet produces even slightly finer masks.

3) SUBEXPERIMENT 3: MSD

Observing per task results of MSD in Table 13, we see that QT-UNet and VT-UNet performs well in Task 1, but falters in others. The in-task pretrained variants of QT-UNet see no change or a slight degradation in performance – though variants pretrained on CT-SSL see an increase in certain tasks. All variants of QT-UNet produce nil-results in Task 7. While the gap between VT-UNet and QT-UNet is not huge, Table 15 shows that VT-UNet performs better overall with the tiny model performing the best.

Qualitatively, the segmentation masks in Figure 17 are decent across all models in the selected tasks, with QT-UNet variant performing slightly better in tasks 6, 9, and 10. The results are good even in the other tasks. The pretrained variant of QT-UNet produces a slightly better mask in tasks 6, 9, and 10. More qualitative results can be seen in Figure 20.

4) ABLATION STUDY

A short ablation study was performed to disentangle the effects of the patch expansion and merging in the depth dimension and our new CA mechanism has on QT-UNet compared to VT-UNet. Variants of VT-UNet and QT-UNet are created for the purpose of this ablation study: VT-UNet-A gains the depth-wise patch merge and expansion, whilst QT-UNet-A drops these operations. An overview of the various models with enabled features can be seen in Table 18. All models in the ablation study are trained from scratch on BraTS, using the same experimental setup as subexperiment 1.

Observing the results in Table 19, we find similar results across all variants, but with a handful of significant differences. The models with depth wise reduction and expansion (VT-UNet-A and QT-UNet) are 33% faster in terms of FLOPs than their counterparts (VT-UNet and QT-UNet-A). Adding depth-wise reduction and expansion to VT-UNet increases model size in terms of parameters by 15%, whereas QT-UNet Tiny, Small, and Base versions increase by 9%, 13%, and 14%, respectively.

Models with the new CA technique (QT-UNet-A and QT-UNet) are 9% faster than models using the old mechanism (VT-UNet and VT-UNet-A). In terms of parameter count, models using the new CA mechanism have a 8% gain. Adding depth reduction and expansion decreases the gain to 3%, 7%, and 6% for the Tiny, Small, and Base variants, respectively.

Overall, QT-UNet has up to 40% fewer FLOPs than VT-UNet at the cost of 23% more parameters. In terms of Dice score, QT-UNet-A-B is tied for first place with VT-UNet-B. However, the Dice scores are pretty comparable overall. All other variants perform worse than VT-UNet-T in terms of average HD.

TABLE 9. BTCV local test split Dice scores (↑) per organ.

Model	Spl	RKid	LKid	Gall	Eso	Liv	Sto	Aor	IVC	Veins	Pan	AG
VT-UNet-T	21.53	53.73	14.50	27.20	0.0	83.75	2.49	24.55	0.0	0.0	1.16	0.0
VT-UNet-S	51.27	76.28	43.23	30.97	0.0	84.65	11.16	45.81	14.41	2.21	22.46	8.69
VT-UNet-B	58.23	71.40	53.60	20.00	0.0	84.71	24.61	37.62	37.62	12.71	16.10	0.0
QT-UNet-T /scratch	62.84	60.70	50.20	24.21	23.70	79.29	24.91	53.03	21.96	39.76	26.06	16.98
QT-UNet-S /scratch	67.56	74.76	57.35	35.54	38.06	82.97	35.70	68.24	46.87	46.47	32.15	34.29
QT-UNet-B /scratch	68.11	75.65	64.95	35.11	41.97	83.80	42.28	68.08	45.95	**49.75**	**38.99**	34.46
QT-UNet-T	**79.19**	78.49	64.49	**40.66**	42.97	**87.77**	36.59	**71.11**	**50.45**	46.69	35.75	32.22
QT-UNet-S	71.31	73.86	59.42	35.97	39.24	84.37	35.28	66.92	44.30	48.64	29.94	32.14
QT-UNet-B	73.37	**80.77**	**65.76**	39.27	**45.54**	83.69	**45.96**	70.63	48.24	52.66	35.71	**36.45**

TABLE 10. BTCV leaderboard Dice scores (↑) per organ.

Model	Spl	RKid	LKid	Gall	Eso	Liv	Sto	Aor	IVC	Veins	Pan	AG
nnUNet [12]	96.70	92.40	**95.70**	81.40	83.20	97.50	92.50	92.80	87.00	83.20	84.90	78.40
nnFormer [13]	90.51	86.25	86.57	70.17	-	96.84	86.83	92.04	-	-	83.35	-
UNETR [11]	97.20	94.20	95.40	82.50	86.40	98.30	94.50	94.80	89.00	85.80	85.20	81.20
Swin-UNETR [14]	**97.60**	**95.80**	95.60	**89.30**	**87.50**	**98.50**	**95.30**	**94.90**	**90.40**	**89.90**	**89.80**	**84.60**

Spl: spleen, RKid: right kidney, LKid: left kidney, Gall: gallbladder, Eso: esophagus, Liv: liver, Sto: stomach, Aor: aorta, IVC: inferior vena cava, Veins: portal and splenic veins, Pan: pancreas, AG: Average of left and right adrenal glands.

TABLE 11. BTCV local test results summary.

Model	#Params ↓	FLOPs ↓	AVG. Dice score ↑
VT-UNet-T	**5.4** M	19.7 G	17.61
VT-UNet-S	11.8 M	38.2 G	30.76
VT-UNet-B	20.8 M	62.6 G	28.46
QT-UNet-T /scratch	6.4 M	**12.7** G	38.51
QT-UNet-S /scratch	14.5 M	23.6 G	50.33
QT-UNet-B /scratch	25.5 M	37.7 G	52.58
QT-UNet-T	6.4 M	**12.7** G	53.50
QT-UNet-S	14.5 M	23.6 G	50.27
QT-UNet-B	25.5 M	37.7 G	**54.96**

TABLE 12. BTCV leaderboard results summary.

Model	#Params ↓	FLOPs ↓	AVG. Dice score ↑
nnUNet [12]	19.07 M	412.65 G	88.80
nnFormer [13]	- M	- G	86.57
UNETR [11]	92.58 M	41.19 G	89.10
Swin-UNETR [14]	61.98 M	394.84 G	**91.80**

DISCUSSION

The experiments and results of the preceding sections will be extensively investigated in order to evaluate our proposed method, and the influence of SSL upon it.

5) RQ1: THE EFFECT OF SSL

The application of SSL to our model gives mixed results depending on the experiment and model variant.This section is divided into two subsections based on the types of SSL used: out-of-task and in-task.

a: EFFECT OF OUT-OF-TASK PRETRAINING

Most tasks see a considerable improvement in performance for QT-UNet models trained out-of-task using the CT-SSL dataset. The model trained on BTCV detects smaller organs better than the baseline. Taking both the BTCV and CT MSD tasks together, we notice that the tiny version benefits the most from pretraining, whereas the small and base variants only show minor improvements.

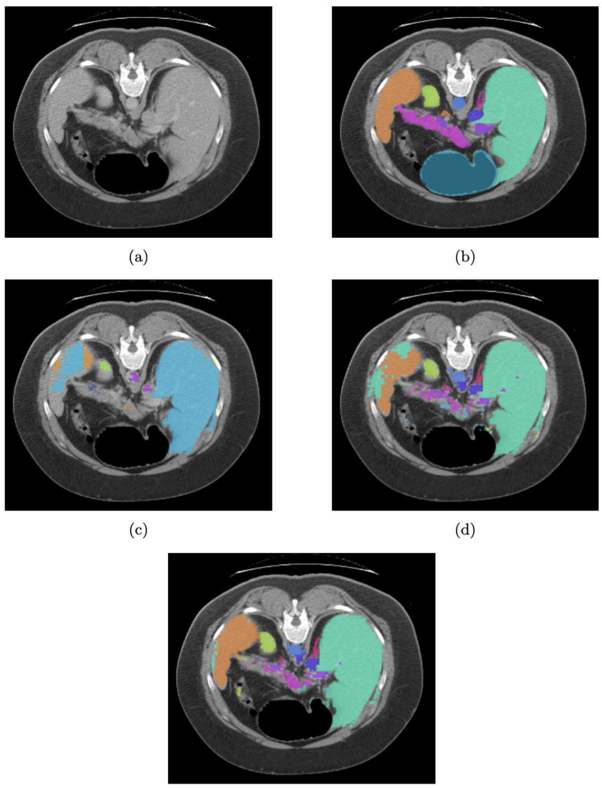


FIGURE 4. Qualitative analysis of various models on one sample from BTCV. (a) Raw image. (b) Ground truth. (c) VT-UNet-B. (d) QT-UNet-B/scratch (e) QT-UNet-B. Light Blue = Stomach, Turquoise = Liver, Orange = Spleen, Light green = Right Kidney, Blue = Aorta, Pink = Pancreas, Dark blue = Inferior Vena Cava, Indigo = Portal and splenic vein, Purple = Adrenal Gland.

We believe this outcome could be attributed to the training procedure and the losses incurred, as illustrated in Figure 5. The graphs reveal that the losses for the base variant never settle in the same manner as the other models, experiencing a sudden increase after 60 epochs. Note that while BYOL loss recovers, rotation and reconstruction loss do not. This leads

TABLE 13. **MSD local test split results per task Dice scores (↑).**

Model	T. 1	T. 2	T. 3	T. 4	T. 5	T. 6	T. 7	T. 8	T. 9	T. 10
VT-UNet-T	78.70	**87.60**	49.06	**85.93**	25.32	40.17	**25.85**	33.38	64.02	12.14
VT-UNet-S	78.20	86.77	47.58	85.18	24.73	**45.78**	23.02	33.11	**69.25**	7.69
VT-UNet-B	**78.61**	87.10	48.57	85.82	26.53	33.38	21.85	33.47	59.40	8.76
QT-UNet-T /scratch	77.37	85.57	36.00	85.45	30.62	26.02	0.0	42.41	49.30	14.11
QT-UNet-S /scratch	77.79	85.81	46.09	82.37	30.90	19.55	0.0	37.07	51.94	13.02
QT-UNet-B /scratch	77.81	86.52	39.31	82.92	32.26	23.21	0.0	33.77	48.10	**15.76**
QT-UNet-T	77.41	85.02	**51.65**	85.23	27.79	27.23	0.0	**45.01**	49.86	14.04
QT-UNet-S	77.68	85.40	37.14	82.19	31.07	19.91	0.0	36.98	53.31	13.52
QT-UNet-B	77.86	86.70	38.56	83.77	**34.08**	23.85	0.0	38.01	56.45	12.87

TABLE 14. **MSD leaderboard results per task Dice scores (↑).**

Model	T. 1	T. 2	T. 3	T. 4	T. 5	T. 6	T. 7	T. 8	T. 9	T. 10
Trans VW [43]	61.14	**93.33**	86.04	**89.53**	81.29	74.54	66.25	68.62	97.35	51.47
Model Genesis [44]	61.14	**93.33**	**86.61**	**89.53**	81.29	74.54	65.86	68.62	97.35	51.47
nnUNet [12]	61.10	93.30	85.86	89.46	**83.11**	73.97	67.21	**69.12**	**99.89**	58.33
Swin-UNETR [14]	66.35	92.62	85.52	89.19	82.40	76.60	**70.71**	68.95	96.99	**59.45**

TABLE 15. **MSD local test results summary.**

Model	AVG. Dice score ↑
VT-UNet-T	**50.22**
VT-UNet-S	50.13
VT-UNet-B	48.35
QT-UNet-T /scratch	44.69
QT-UNet-S /scratch	44.45
QT-UNet-B /scratch	43.97
QT-UNet-T	46.32
QT-UNet-S	43.72
QT-UNet-B	45.22

TABLE 16. **MSD leaderboard results summary.**

Model	AVG. Dice score ↑
Trans VW [43]	76.96
Model Genesis [44]	76.97
nnUNet [12]	78.14
Swin-UNETR [14]	**78.88**

us to theorise that the base model somehow collapsed during pre-training. This is a risk that the authors of BYOL [20] warn of, claiming that a collapse of BYOL where the model outputs only zero-vectors as projections since that also would provide a minima of loss.

It is also possible that the interaction between the SSL heads causes the collapse. We couldn't discover any study on the usage of BYOL in this sort of multi-head strategy, thus its interaction with the other approaches is unknown.

b: EFFECT OF IN-TASK PRETRAINING

The in-task trained MRI tasks also provide interesting results. In BraTS2021, pretraining improved QT-UNet accuracy, lowering HD while maintaining the Dice score. This improvement can also be observed in the qualitative results.

Pre-trained and standard models differ only slightly in in-task trained MSD tasks 1, 4, 5, and 6. Most tasks show negligible Dice score changes, with some models increasing and others decreasing. A larger out-of-task dataset might have helped the SSL technique to extract stronger representations from the limited data.

TABLE 17. **Qualitative results of selected MSD tasks.**

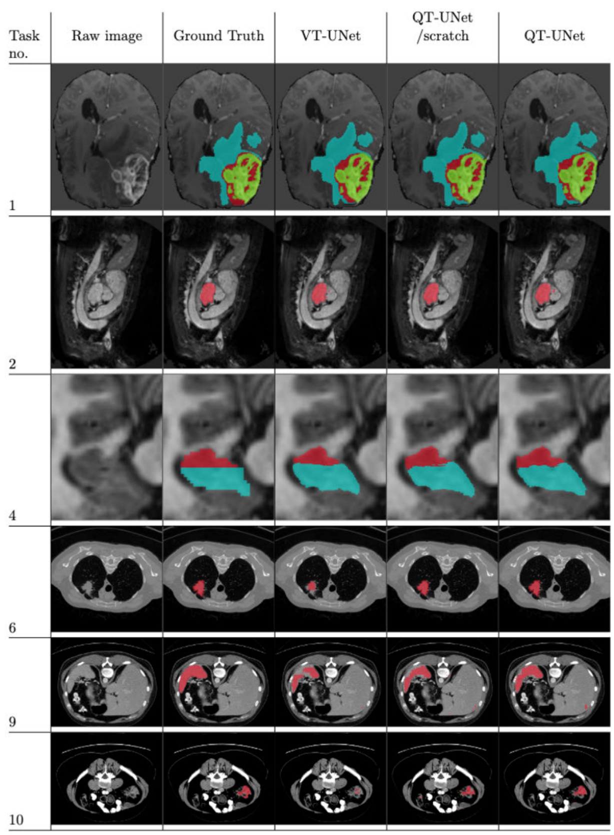


TABLE 18. **Overview of ablation model features.**

Model	Depth-wise	New CA module
VT-UNet	X	X
VT-UNet-A	✓	X
QT-UNet-A	X	✓
QT-UNet	✓	✓

TABLE 19. Ablation study of various models on BraTS2021 data.

Model	Params ↓	FLOPs ↓	Dice score ↑				Hausdorff Distance ↓			
			ET	TC	WT	Avg.	ET	TC	WT	Avg.
VT-UNet-T	**5.4** M	52 G	85.71	**88.40**	91.94	88.68	4.12	4.73	5.12	**4.69**
VT-UNet-S	11.8 M	100.8 G	**86.58**	88.01	91.85	**88.82**	4.16	**4.68**	5.35	5.10
VT-UNet-B	20.8 M	165 G	86.11	87.88	91.89	88.63	4.22	5.13	**4.53**	4.71
VT-UNet-A-T	6.2 M	35.0 G	85.16	88.00	91.64	88.27	**4.04**	4.98	4.93	4.74
VT-UNet-A-S	13.5 M	67 G	85.62	87.62	91.62	88.29	4.25	5.08	5.24	4.92
VT-UNet-A-B	23.9 M	108.7 G	86.07	87.62	91.60	88.43	4.23	4.93	4.82	4.74
QT-UNet-A-T /scratch	5.9 M	47.7 G	86.15	87.95	92.01	88.70	4.77	5.98	6.18	5.69
QT-UNet-A-S /scratch	12.8 M	91.2 G	85.66	87.96	92.17	88.60	4.68	4.93	5.91	4.96
QT-UNet-A-B /scratch	22.4 M	147.8 G	86.36	87.86	**92.24**	**88.82**	4.45	5.88	7.01	5.97
QT-UNet-T /scratch	6.4 M	**32.5** G	85.38	87.00	92.06	88.15	4.32	6.24	6.48	5.79
QT-UNet-S /scratch	14.5 M	61.3 G	85.90	87.60	92.20	88.56	4.18	5.17	5.95	5.39
QT-UNet-B /scratch	25.5 M	98.5 G	86.27	87.61	92.19	88.69	4.23	5.14	5.21	4.92

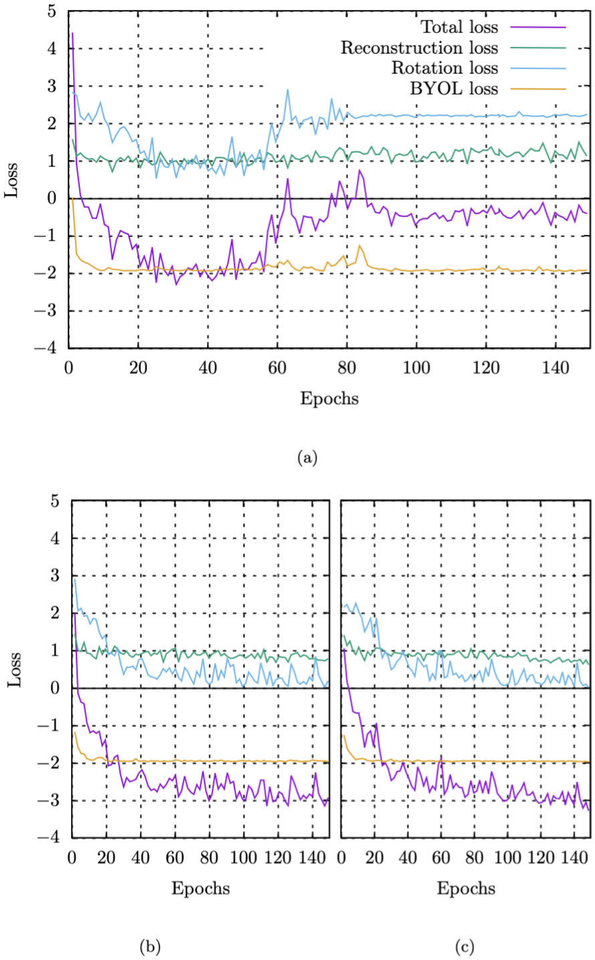


FIGURE 5. Loss curves for CT-SSL pretraining. (a) Loss curves for QT-UNet-B (b) Loss curves for QT-UNet-S (c) Loss curves for QT-UNet-T.

6) RQ2: ENCODER-DECODER CROSS-ATTENTION

The ablation study in Table 19 shows that the new CA module decreases computational burden by 8.27%, 9.51%, and 10.42% for the tiny, small, and base variants, respectively, while increasing parameters by 9.26%, 8.47%, and 7.69%.

The Dice scores are mixed, with some models experiencing a slight improvement while others see a minor reduction in performance. The HD, on the other hand, is adversely impacted across all versions. There might be various explanations for this. One of them is that the new CA decoder module is simply smaller than its counterpart due to its single stream architecture.

QT-UNet consistently outperforms VT-UNet on BTCV dataset by an average of 18 Dice points across all variants. QT-UNet segmented the smallest organs better than VT-UNet, suggesting that the new CA architecture was better at querying the encoder for the spatial location of these organs. It can be clearly observed that QT-UNet effectively differentiates the right kidney from the left.

However, in MSD tasks 5, 8, and 10, QT-UNet outperforms VT-UNet, whereas VT-UNet outperforms QT-UNet in all other tasks. QT-UNet's depth-wise merge and expansion and the new CA mechanism could be responsible for this. These tasks have few target classes, typically one or two. Task 9 is essentially the same task as segmenting the spleen organ in BTCV at which QT-UNet outperforms VT-UNet in BTCV. This indicates that the new CA mechanism is affected by the number of target classes, performing better in a "target rich environment."

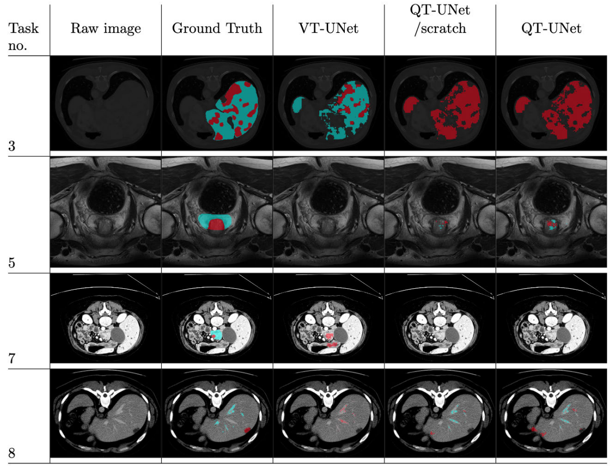
QT-UNet failed in MSD task 7, where all variants scored zero. These runs collapse midway through training, with training loss decreasing and validation loss increasing. The very small targets could be collapsing the model, predicting background everywhere.

Overall, the new CA module has a superior speed-to-parameter trade-off, lowering computational burden. Dice score impacts tasks with many target classes positively but negatively with few target classes. It can also be observed that HD also decreases with BraTS data.

V. CONCLUSION

We found that the overall effect of self-supervised pre-training varies significantly depending on whether in-task or out-of-task data is used. SSL utilising out-of-task data improved model performance significantly in terms of Dice score. Training with in-task data improved the Hausdorff Distance on BraTS dataset, but had negligible impact when tested with the Dice scores overall.

TABLE 20. Additional MSD qualitative results.



Furthermore, we found that the updated CA mechanism in QT-UNet achieved a better speed-to-performance trade-off compared to VT-UNet-B on BraTS, trading a 7.69% gain in parameters for a 10.42% drop in FLOPs with negligible impact on Dice score. The new mechanism boosted the average Dice score 18 points higher on BTCV data. It is also better at detecting small organs. However, the new technique is less accurate in terms of HD than the VT-UNet approach. The new mechanism is also observed to perform better in tasks with many target classes.

While some of our experiments yielded mixed results, our overall method showed great promise, being 40% faster than the baseline VT-UNet and having validated the effects of both our new CA mechanism and the SSL setup. We believe that the performance gaps discovered could be mitigated with the further development.

APPENDIX
ADDITIONAL MSD QUALITATIVE SAMPLES

Additional qualitative results from MSD task omitted from the results section due to space constraints can be seen in Table 20.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol. (NAACL-HLT)*, vol. 1, Minneapolis, MN, USA, J. Burstein, C. Doran, and T. Solorio, Eds. New York, NY, USA: Association for Computational Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.

[3] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, May 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[5] N. Kitaev, L. Kaiser, and A. Levskaya, "ReFormer: The efficient transformer," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, 2020. [Online]. Available: https://openreview.net/forum?id=rkgNKkHtvB

[6] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020, *arXiv:2006.04768*.

[7] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," 2021, *arXiv:2108.09084*.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[9] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-Unet: Unet-like pure transformer for medical image segmentation," 2021, *arXiv:2105.05537*.

[10] H. Peiris, M. Hayat, Z. Chen, G. Egan, and M. Harandi, "A robust volumetric transformer for accurate 3D tumor segmentation," 2021, *arXiv:2111.13300*.

[11] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "UNETR: Transformers for 3D medical image segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1748–1758.

[12] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "NnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Dec. 2020, doi: 10.1038/s41592-020-01008-z.

[13] H.-Y. Zhou, J. Guo, Y. Zhang, L. Yu, L. Wang, and Y. Yu, "NnFormer: Interleaved transformer for volumetric segmentation," 2021, *arXiv:2109.03201*.

[14] Y. Tang, D. Yang, W. Li, H. R. Roth, B. Landman, D. Xu, V. Nath, and A. Hatamizadeh, "Self-supervised pre-training of Swin Transformers for 3D medical image analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20698–20708.

[15] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video Swin Transformer," 2021, *arXiv:2106.13230*.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, in Lecture Notes in Computer Science, vol. 9351, Munich, Germany, N. Navab, J. Hornegger, W. M. W. III, and A. F. Frangi, Eds. Cham, Switzerland: Springer, Oct. 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[17] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Red Hook, NY, USA: Curran Associates, 2021, pp. 9355–9366. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/4e0928de075538c593fbdabb0c5ef2c3-Paper.pdf

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," 2015, *arXiv:1502.01852*.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 119, H. D. III and A. Singh, Eds., Jul. 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[20] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 21271–21284. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

[21] *PyTorch Lightning (1.4)*. William Falcon The PyTorch Lightning Team. Accessed: Jan. 31, 2022. [Online]. Available: https://github.com/PyTorchLightning/pytorch-lightning

[22] A. H. Håversen, "QT-UNet: A self-querying all-transformer U-Net for 2D and 3D segmentation augmented by self-supervised learning," M.S. thesis, Dept. Comput. Sci., NTNU, Trøndelag, Norway, 2022. [Online]. Available: https://hdl.handle.net/11250/3019918

[23] *Anaconda Software Distribution (2.4.0)*. Anaconda Inc. Accessed: Jan. 31, 2022. [Online]. Available: https://anaconda.com/

[24] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Dec. 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] W. Falcon and K. Cho, "A framework for contrastive self-supervised learning and designing a new approach," 2020, *arXiv:2009.00104*.

[26] M. J. Cardoso et al., "MONAI: An open-source framework for deep learning in healthcare," 2022, *arXiv:2211.02701*.

[27] M. Själander, M. Jahre, G. Tufte, and N. Reissmann, "EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure," 2019, *arXiv:1912.05848*.

[28] T. Sørenson, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons* (Biologiske Skrifter), E. Munksgaard, Ed. Copenhagen, Denmark: I kommisjon hos E. Mungsgaard, 1948. [Online]. Available: https://books.google.no/books?id=rpS8GAAACAAJ

[29] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*. Berlin, Germany: Springer, 1998, doi: 10.1007/978-3-642-02431-3.

[30] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, Dec. 2013, doi: 10.1007/s10278-013-9622-7.

[31] H. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers. (2015). *A New 2.5 D Representation for Lymph Node Detection in CT*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/0gAtAQ

[32] C. K. Smith. (2015). *Data From CT_Colonography*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/DQE2

[33] S. Desai, A. Baghal, T. Wongsurawat, S. Al-Shukri, K. Gates, P. Farmer, M. Rutherford, G. D. Blake, T. Nolan, T. Powell, K. Sexton, W. Bennett, and F. Prior. (2020). *Chest Imaging With Clinical and Genomic Correlates Representing a Rural COVID-19 Positive Population*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/C5IvB

[34] E. Tsai et al. (2020). *Medical Imaging Data Resource Center—RSNA International COVID Radiology Database Release 1A—Chest CT COVID+ (MIDRC-RICORD-1A)*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/DoDTB

[35] E. B. Tsai et al. (2021). *Medical Imaging Data Resource Center (MIDRC)—RSNA International COVID Open Research Database (RICORD) Release 1B—Chest CT COVID—(MIDRC-RICORD-1B)*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/K4DTB

[36] A. A. Yorke, G. C. McDonald, D. Solis, and T. Guerrero. (2019). *Pelvic Reference Data*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/tQGJAw

[37] T. Tong and M. Li. (2022). *Abdominal or Pelvic Enhanced CT Images Within 10 Days Before Surgery of 230 Patients With Stage II Colorectal Cancer*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/3wL7Bg

[38] S. G. Armato III et al. (2015). *Data From LIDC-IDRI*. [Online]. Available: https://wiki.cancerimagingarchive.net/x/rgAe

[39] Fair Computer Vision Team. (2021). *FacebookResearch/FVCore: Collection Common Code That's Shared Among Different Research*. [Online]. Available: https://github.com/facebookresearch/fvcore

[40] H. M. Luu and S.-H. Park, "Extending nn-UNet for brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham, Switzerland: Springer, 1007, pp. 173–186.

[41] M. M. R. Siddiquee and A. Myronenko, "Redundancy reduction in semantic segmentation of 3D brain tumor MRIs," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds. Cham, Switzerland: Springer, 1007, pp. 163–172.

[42] R. A. Zeineldin, M. E. Karar, F. Mathis-Ullrich, and O. Burgert, "Ensemble CNN networks for GBM tumors segmentation using multi-parametric MRI," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds. Cham, Switzerland: Springer, 1007, pp. 473–483.

[43] F. Haghighi, M. R. H. Taher, Z. Zhou, M. B. Gotway, and J. Liang, "Transferable visual words: Exploiting the semantics of anatomical patterns for self-supervised learning," *IEEE Trans. Med. Imag.*, vol. 40, no. 10, pp. 2857–2868, Oct. 2021.

[44] Z. Zhou, V. Sodha, M. M. R. Siddiquee, R. Feng, N. Tajbakhsh, M. B. Gotway, and J. Liang, "Models genesis: Generic autodidactic models for 3D medical image analysis," in *Proc. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham, Switzerland: Springer, 2019, pp. 384–393.

ANDREAS HAMMER HÅVERSEN was born in Lørenskog, Akershus, Norway, in 1998. He received the M.S. degree in computer science with a specialization in artificial intelligence from Norwegian University of Science and Technology, in 2022. Since 2022, he has been a Software Engineer with Bekk Consulting AS, Oslo, Norway, developing software for clients in Norwegian IT Sector.

DURGA PRASAD BAVIRISETTI received the Ph.D. degree in signal and image processing from VIT, in 2016. He completed postdoctoral research with Shanghai Jiao Tong University. He was an Algorithm Expert with the Innovation Center, Alibaba Group, and was a Visiting Researcher with UBC Okanagan and the University of Warsaw. He is currently a Researcher and an Algorithm Developer specializing in machine learning, deep learning, and computer vision. He is a Researcher with NTNU, Norway, working on machine vision for autonomous driving and medical image computing.

GABRIEL HANSSEN KISS received the Diploma degree in computer science engineering from the Technical University of Cluj-Napoca, Romania, and the Ph.D. degree in engineering from KU Leuven, Belgium. He is currently an Associate Professor with the Computer Science Department, NTNU, Trondheim, and a Senior Engineer with the Operating Room of the Future, St. Olav's University Hospital. His main area of expertise is related to vision computing, medical image processing and visualization, digital twins, and ultrasound technology. Special interests include extended reality, volumetric data visualization, image registration, and fusion for ultrasound-related applications.

FRANK LINDSETH received the B.Sc. degree in engineering from BIH, Bergen, Norway, and the M.Sc. degree in mathematical science and the Ph.D. degree in computer science from Norwegian University of Science and Technology (NTNU). He is currently a Professor with the Department of Computer Technology and Informatics, Faculty of Information Technology and Electrical Engineering, NTNU. He is also a Senior Research Scientist with SINTEF Medical Technology, where he works in the field of image-guided interventions/surgical navigation. His areas of expertise include image-guided surgery, navigation and tracking technology, medical image processing and analysis, medical visualization, computer graphics, and augmented reality. He is also skilled in human–computer interaction, GUI design, open-source software development, and project management.