mCNA: Multi-Clone Communication Neural Architecture Search for 3D Medical Image Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Volumetric medical image segmentation requires precise delineation of anatomical structures under substantial variations in organ size, shape, and appearance, while being constrained by high memory and computational costs of 3D processing. Neural Architecture Search (NAS) provides a principled approach for automated architecture design, but directly searching large 3D segmentation networks is prohibitively expensive due to the combinatorial complexity of jointly optimizing cell operations, skip connections, network topology, and model capacity under dense volumetric representations. To address this challenge, we propose Multi-Clone Communication Neural Architecture (mCNA), a search-efficient 3D NAS framework that decomposes 3D architecture optimization into compact cell search and clonable macro-level expansion. Instead of searching a complete large-scale U-Net topology, mCNA first discovers an optimized 3D U-Net backbone and then constructs a larger architecture by replicating the searched unit with Communication Units for inter-clone feature exchange and cooperative refinement. To avoid costly discrete skip-topology exploration, we introduce a Path Attention Module that performs differentiable adaptive fusion of multi-scale encoder features. Furthermore, we incorporate structure-preserving wavelet-based downsampling operations with pre-optimized and fixed filters into the search space to retain fine-grained volumetric information during downsampling. Experiments on Synapse and the Medical Segmentation Decathlon benchmark demonstrate that mCNA achieves the best overall performance among the evaluated methods under aligned preprocessing, training, and evaluation protocols. Extensive ablation studies verify the effectiveness of the proposed cell search, clone communication, adaptive skip fusion, and wavelet-based downsampling components. The searched architecture can also be transferred across datasets without repeating the micro-cell architecture search, demonstrating its reuse across diverse segmentation tasks.

1 Introduction

Reliable 3D abdominal CT segmentation plays an essential role in quantitative analysis, treatment planning, and surgical navigation. However, accurate volumetric segmentation remains challenging due to large variations in organ size, shape, and spatial configuration, as well as severe class imbalance for small and boundary-sensitive structures. While recent CNN, Transformer, and state-space-model-based backbones have improved segmentation accuracy, most architectures and skip-fusion strategies are still manually designed and may not generalize optimally across datasets. Neural Architecture Search (NAS) (Elsken et al., 2019) provides a systematic framework for automatically discovering effective network architectures. However, extending NAS from 2D recognition tasks to 3D medical segmentation remains highly challenging. A 3D supernet must preserve multi-resolution volumetric features across numerous candidate operators and connections, causing substantial GPU memory and computation overhead. Dense 3D validation mostly relies on sliding-window inference, making frequent architecture evaluation prohibitively expensive. Such challenges make it impractical to jointly search complex macro-level topologies and micro-level cell architectures within a large-scale 3D supernet.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A key challenge in NAS is therefore not only designing powerful search spaces, but also determining which architectural factors should be searched and which can be constructed efficiently. We explicitly prioritize what to search and how to reduce the associated complexity. Most segmentation architectures (Fourure et al., 2017; Liu et al., 2019; Ronneberger et al., 2015; Weng et al., 2019) adopt U-Net–style designs to achieve strong performance. For example, AutoDeepLab constructs a level-based U-Net supernet whose search space grows exponentially with the number of levels L and depth D. Jointly searching both network-level topology and cell-level operations introduces substantial computational overhead and leads to inefficient optimization. AutoDeepLab continuously optimizes transitions among multiple resolutions during search and subsequently extracts a discrete network-level path through post-search decoding. In contrast, mCNA avoids explicit network-level topology search and instead focuses the discrete architecture search on compact cell structures, while constructing the macro architecture through clone-based expansion.

We propose Multi-Clone Communication Neural Architecture (mCNA), a clonable architecture that reduces the complexity of 3D NAS. Instead of searching a large-scale heterogeneous network containing multiple independent components, mCNA searches a single optimized U-Net architecture and constructs the macro architecture through architecture-preserving replication. Specifically, mCNA consists of multiple cloneable U-Net units that share the same searched structure, where learnable Communication Units enable information exchange between neighboring clones for cooperative volumetric refinement. This design significantly reduces the search space while allowing the final network to increase representational capacity through efficient clone-based expansion.

Note that mCNA is built upon a differentiable optimization framework similar to DARTS (Liu et al., 2018), enabling efficient gradient-based search compared to reinforcement learning or evolutionary NAS methods. Unlike prior DARTS-based methods that mainly focus on cell-level design, mCNA combines cell-level architecture search with a computationally tractable macro-level expansion strategy and differentiable skip-path weighting. To enable adaptive multi-scale feature fusion without introducing an additional discrete topology search space, we propose the Path Attention Module (PAM), which performs differentiable soft weighting and fusion of candidate encoder features at each decoder stage. Unlike prior methods such as AutoDeepLab and HiNAS, which restrict each node to a limited number of candidate inputs and typically decode a single dominant path (*e.g.*, through Viterbi decoding), PAM aggregates complementary encoder features from multiple resolutions. This design avoids exhaustive skip-path enumeration while providing richer multi-scale representations for segmentation.

We address the irreversible information loss caused by conventional downsampling operations. Standard pooling methods (*e.g.*, stride-2 max pooling) discard local spatial responses, making fine-grained details difficult to recover. To mitigate this issue, we extend Learnable Discrete Wavelet Pooling (LDW) (Wang et al., 2021a) to volumetric feature decomposition. Reversibility and orthogonality constraints preserve complementary low- and high-frequency information while reducing redundancy, thereby retaining both anatomical context and structural details during downsampling. We denote this formulation as Orth. LDW. The wavelet filters are optimized separately and fixed during architecture search and network retraining. Moreover, Orth. LDW remains computationally efficient because its 2D/3D convolutions can be decomposed into sequential 1D operations.

Our proposed mCNA is simple, efficient, and effective. The differentiable micro-cell search can be completed within half a day on a single A100 GPU. Extensive experiments on both 3D datasets (*e.g.*, Synapse and MSD) and 2D benchmarks (*e.g.*, CHAOS, BAGLS, and KVASIR) demonstrate strong segmentation performance across diverse medical imaging tasks. The main contributions of this paper are summarized as follows:

1. **Clonable NAS formulation**: We propose mCNA, a multi-clone communication U-Net framework reducing the NAS search space by decoupling cell and macro-architecture design: a single optimized U-Net unit is searched once and then cloned with shared searched architecture to construct the global architecture.
2. **Adaptive skip-path aggregation**: We propose a Path Attention Module (PAM) that performs differentiable, input-adaptive weighting and fusion of multi-scale skip features, avoiding explicit discrete skip-topology search.
3. **Information-preserving downsampling**: We extend LDW pooling to mitigate the irreversibility of conventional pooling, preserving fine-grained spatial details while maintaining computational efficiency.

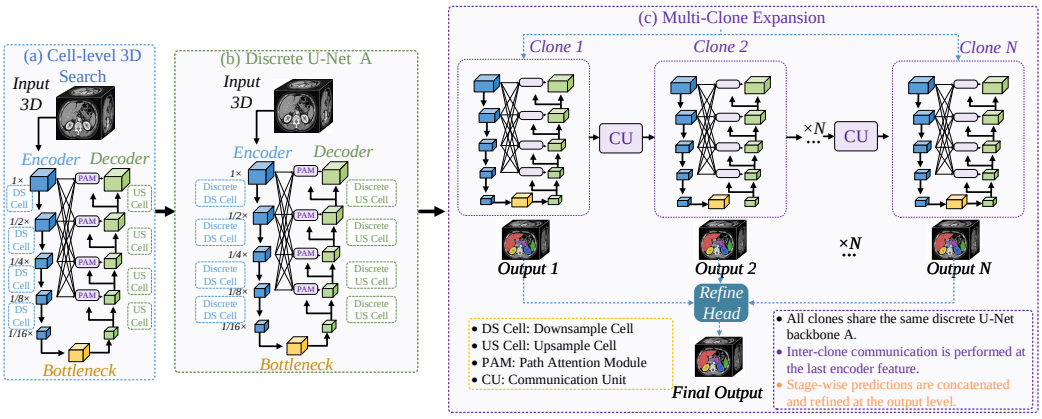

Figure 1: Overall framework of the proposed mCNA. (a) A 3D U-shaped supernet is searched to optimize the downsampling and upsampling cells. (b) The searched supernet is discretized into a reusable U-Net backbone $\mathcal{A}$. (c) The discrete backbone is expanded into multiple communicating clones, each equipped with a Communication Unit (CU).

4. **Strong cross-task performance**: mCNA achieves the highest mean Dice and lowest HD95 on Synapse, as well as the highest 10-task macro-average Dice and the highest task-level Dice on nine of the ten MSD tasks under the aligned evaluation protocol.

2 Related Work

3D Medical Image Segmentation Backbones. Convolutional encoder–decoder networks dominate 3D medical image segmentation for efficiency, stable optimization, and multi-resolution fusion. Methods like 3D U-Net (Çiçek et al., 2016) use U-shaped skip-connected architectures for volumetric prediction, while SegResNet (Myronenko, 2018) enhances features with residual blocks. Recent CNN designs (*e.g.*, 3D UX-Net (Lee et al., 2022), MedNeXt (Roy et al., 2023)) further improve convolutions and training. Transformer models (UNETR (Hatamizadeh et al., 2022), Swin-UNETR (Hatamizadeh et al., 2021)) add self-attention for long-range 3D dependencies. State-space models (U-Mamba (Ma et al., 2024), SegMamba (Xing et al., 2024), SegMamba-V2 (Xing et al., 2025), MM-UNet (Xie et al., 2026), SegResMamba (Das et al., 2026)) target efficient global context. Despite strong performance, architectures, skip fusion, and capacity are largely hand-crafted, limiting transferability across datasets and compute constraints.

NAS for Medical Image Segmentation. NAS is widely used in vision (Elsken et al., 2019; Liu et al., 2018) to automate design. For medical segmentation, it often searches U-shaped cells, operators, or paths to replace hand-crafted encoder–decoders (Liu et al., 2019; Yan et al., 2020; Wei et al., 2026). Extending NAS to 3D is harder than 2D, as 3D supernets require dense multi-resolution volumes and far higher memory and compute during search/validation. Resolution changes and skip connections also impose topology and shape constraints. Thus, prior 3D NAS restricts search space, uses memory-aware methods, or decomposes into cell/topology-level tasks. DiNTS (He et al., 2021) performs differentiable topology search with memory efficiency. Others explore adaptive, multi-scale, or topology/path search (Huang et al., 2022; Peng et al., 2022; Yu et al., 2020; Zhang et al., 2020a; Benedykciuk et al., 2026). Unlike these, mCNA avoids large 3D supernets: it first learns a reusable backbone via cell search on a 3D U-Net, then expands it through multi-clone construction and inter-clone communication.

Path Search and Multi-scale Feature Preservation. Beyond cell-level search, path topology and skip connections are key in encoder–decoder segmentation networks. Auto-DeepLab (Liu et al., 2019) and HiNAS (Zhang et al., 2020a) perform network-level/hierarchical path search for cross-resolution feature transitions. Decoder-guided skip fusion (*e.g.*, UDTransNet's DRA module (Wang et al., 2024)) recalibrates skip features using decoder context to reduce the semantic gap. How-

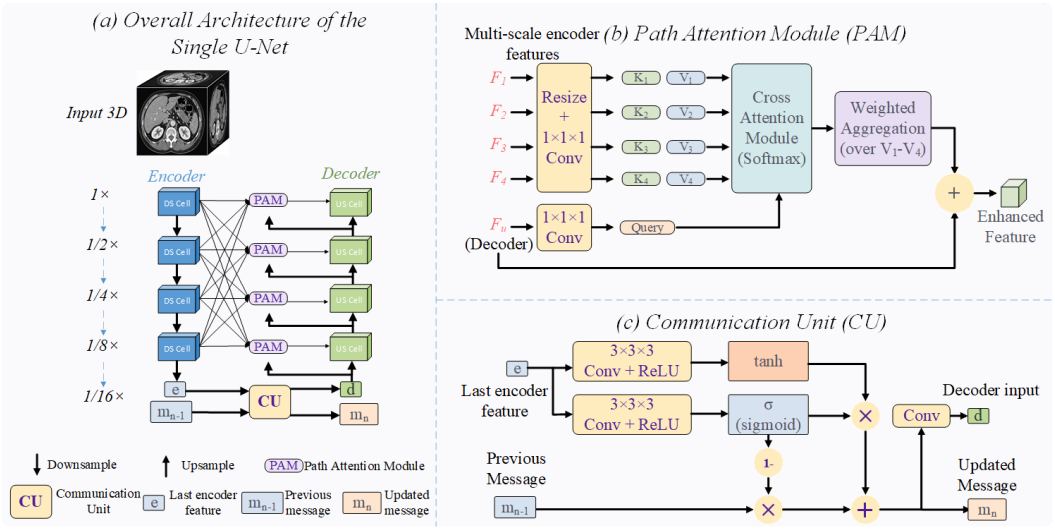

Figure 2: Illustration of the discrete U-Net backbone and the proposed feature-fusion modules. (a) Single U-Net architecture with PAM-based multi-scale skip fusion and CU-based inter-clone communication. (b) Path Attention Module (PAM). (c) Details of Communication Unit (CU).

ever, these refine fixed skips and do not address the combinatorial cost of selecting multi-scale skip paths, especially in 3D NAS. Moreover, preserving features during downsampling is crucial for small organs, thin structures, and low-contrast boundaries; wavelet pooling decomposes features into frequency subbands and helps retain information (Fujieda et al., 2018; Williams & Li, 2018; Hao et al., 2026).

3 METHOD

Overview: We propose mCNA, a 3D multi-clone communication NAS U-Net framework for volumetric segmentation. Let C, D, H, and W denote the number of input channels, depth, height, and width, respectively. Given an input volume $X \in \mathbb{R}^{C\times D\times H\times W}$, the network predicts voxel-wise logits $\hat{Y} \in \mathbb{R}^{K\times D\times H\times W}$ over K semantic classes. As shown in Fig. 1, mCNA consists of three stages. First, we perform cell-level architecture search within a single U-shaped supernet to optimize the operators and internal connections of the downsampling and upsampling cells. Second, the searched supernet is discretized into a reusable 3D U-Net backbone $\mathcal{A}$. Finally, $\mathcal{A}$ is cloned into N communicating U-Nets with a CU at each bottleneck. PAM and Orth. LDW provide multi-scale skip fusion and structure-preserving downsampling, respectively, while a lightweight refinement head fuses clone predictions. The complete procedure is detailed in Appendix A.

3.1 MULTI-CLONE COMMUNICATION 3D U-NETS

The searched backbone $\mathcal{A}$ is a cell-level 3D U-Net composed of L encoding nodes and L decoding nodes. Each encoding node receives the feature from the previous encoding node and produces an encoder feature that can serve as a skip candidate for decoding. The last encoding node produces a bottleneck feature, which provides the deepest semantic representation of the input volume. For each decoding node, we consider both the feature from the previous decoding node and encoder features from multiple depths. The multi-scale encoder features are aggregated by PAM to form the skip feature for each decoder node, as detailed in Sec. 3.5.

To construct the final network, $\mathcal{A}$ is cloned into N U-Nets with the same discrete architecture. Each clone contains a CU; the first uses $m_0 = 0$, while subsequent clones receive the message propagated from the preceding clone. Their predictions are concatenated and processed by an output refinement head. Fig. 2(a) shows the placement of PAM and CU.

3.2 GATED CU FOR INTER-CLONE COMMUNICATION

As shown in Fig. 2(c), CU performs gated message updating between adjacent clones and injects the updated message into the current decoder. Let $e_n \in \mathbb{R}^{C_b \times D_b \times H_b \times W_b}$ denote the bottleneck feature of the n-th clone, where C_b is the number of bottleneck channels and D_b, H_b, and W_b denote the bottleneck depth, height, and width, respectively, and let m_{n-1} be the message map propagated from the $(n-1)$-th clone (initialized as zeros for $n = 1$). We transform e_n by a $3 \times 3 \times 3$ convolution followed by ReLU and then regularize it with $\tanh(\cdot)$:

$$e'_n = \tanh\left(f^{\mathrm{ReLU}}_{3\times3\times3}(e_n)\right), \tag{1}$$

where $f^{\mathrm{ReLU}}_{3\times3\times3}(\cdot)$ denotes a $3 \times 3 \times 3$ convolution followed by a ReLU activation. To estimate the reliability of the bottleneck feature, we compute an updating confidence (gate) c_n as

$$c_n = \sigma\left(g^{\mathrm{ReLU}}_{3\times3\times3}(e_n)\right), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and $g^{\mathrm{ReLU}}_{3\times3\times3}(\cdot)$ is another $3 \times 3 \times 3$ convolution followed by ReLU. Similar to the update gate in GRU (Cho et al., 2014), c_n controls how much information from the clone should be written into the propagated message. The message is then updated as

$$m_n = (1 - c_n) \odot m_{n-1} + c_n \odot e'_n, \tag{3}$$

where $\odot$ denotes element-wise multiplication. The message injected into the decoder of the n-th clone is obtained by

$$d_n = h_{3\times3\times3}(m_n), \tag{4}$$

where $h_{3\times3\times3}(\cdot)$ is a $3 \times 3 \times 3$ convolution. Unlike GRU-style recurrent states, m_n is a residual 3D feature map preserving spatial context across clones. The injected feature d_n serves as the decoder-side input F_u to the deepest US-Cell of the n-th clone, providing communication-aware bottleneck information for inter-clone refinement.

3.3 SEARCH SPACE AND CELL ARCHITECTURE SEARCH

We perform architecture search on a cell-level single 3D U-shaped supernet. The search focuses on the micro-architecture of downsampling cells (DS-Cells) and upsampling cells (US-Cells), including their internal connections and primitive operations. Each DS-Cell receives one input feature, increases the channel dimension from C to $2C$, and reduces the spatial resolution from (D, H, W) to $(D/2, H/2, W/2)$. Each US-Cell receives two scale-compatible input features: the current decoder feature F_u and the PAM-enhanced feature P_u. It reduces the channel dimension from C to $C/2$ and enlarges the spatial resolution from (D, H, W) to $(2D, 2H, 2W)$. We use three intermediate nodes to construct each cell, as illustrated in Fig. 3. The complete primitive operation set $\mathcal{O}$ is provided in Appendix B.1. For each DS-Cell, the input feature is first projected into two initial feature maps with the same spatial size using $1 \times 1 \times 1$ convolutions followed by nonlinear activation. These initial features are then fed into the cell DAG, where mixed operations are applied along searchable edges to generate the downsampled output feature.

Following DARTS (Liu et al., 2018), each searchable edge in a DS/US cell is represented as a mixed operation over the primitive set $\mathcal{O}$. Given an input feature x, the mixed operation on edge (i, j) is defined as

$$\bar{o}^{(i,j)}(x) = \sum_{o\in\mathcal{O}} \mathrm{softmax}\Big(\alpha^{(i,j)}\Big)_o \, o(x), \tag{5}$$

where $\alpha^{(i,j)}$ denotes the architecture weights over candidate operations on edge (i, j). During search, both the operation weights and architecture weights are optimized using differentiable architecture search.

PAM is used as a differentiable soft skip-path weighting module for encoder–decoder feature fusion. It keeps multi-scale skip candidates adaptive without introducing extra discrete skip-topology variables, so the discrete architecture search remains focused on the DS/US cell structures. For downsampling, we also include our proposed LDW pooling operators in the primitive set as structure-preserving candidates. The detailed formulation of Orth. LDW is described in the "Orthogonality- and Reversibility-Constrained Learnable Discrete Wavelet Pooling" subsection. After search, each intermediate node retains its two strongest incoming edges and the highest-weighted operation on each edge. The resulting 3-meta-node DS/US cells are shown in Fig. 4 in Appendix A.

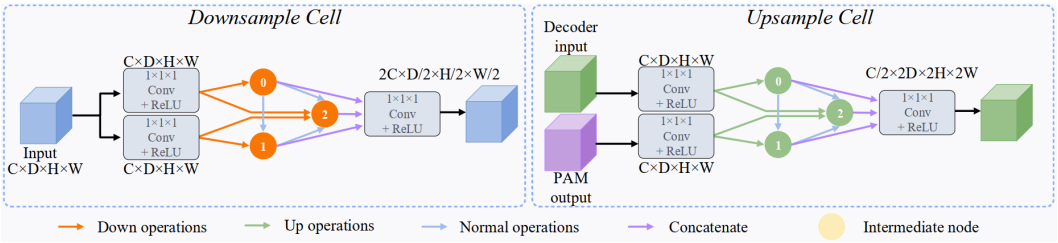

Figure 3: Left: One-input downsampling cell. Right: Two-input upsampling cell.

3.4 Orthogonality- and Reversibility-Constrained Learnable Discrete Wavelet Pooling

Conventional max or average pooling operations are generally irreversible because they compress local spatial responses during downsampling. This information loss can be harmful for small organs, thin anatomical structures, and boundary-sensitive regions in volumetric medical images. To improve structure preservation, we propose the volumetric LDW pooling in the downsampling search space. The LDW pooling decomposes 3D feature volumes using learnable low-pass and high-pass filters. The decomposition is performed separably along the depth, height, and width axes, producing eight volumetric subband responses: $\{LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH\}$, where a low-pass ($L$) or high-pass ($H$) filtering operation is applied along one spatial axis. Thus, the resulting subbands capture different combinations of volumetric anatomical context and boundary-sensitive detail responses.

Rather than using LDW as a fixed downsampling layer, we expose each volumetric subband as a primitive operation in the downsampling search space. This allows the architecture search process to select task-relevant structure-preserving downsampling patterns jointly with other candidate operations. For example, low-frequency subbands can preserve coarse anatomical context, high-frequency subbands can emphasize local structural changes, and mixed subbands can encode directional volumetric variations. In this way, LDW provides a searchable set of wavelet-derived, structure-preserving downsampling candidates instead of a manually fixed pooling rule. The low-pass and high-pass filters are first learned in a separate filter-learning stage using normalization, reversibility, and orthogonality constraints. Once learned, their coefficients remain fixed during both differentiable architecture search and network retraining. During architecture search, the fixed LDW subband operations are included as candidates, while the architecture weights governing their selection are optimized. The filter-learning objective and regularization terms are provided in Appendix C.

3.5 Path Attention Module (PAM) for Soft Path Weighting

Explicitly searching encoder–decoder skip paths can greatly enlarge the macro-level search space, while hard selection of only a few paths may discard useful multi-scale information. We thus propose **Path Attention Module (PAM)** to relax discrete skip-path selection into differentiable soft skip-path weighting. For each decoder level, PAM aggregates candidate encoder features from multiple depths and produces a skip feature that is scale-compatible with the decoder feature. As illustrated in Fig. 2(b), let $\{F_i\}_{i=1}^{M}$ denote the candidate encoder features from different depths, and let F_u denote the current decoder feature. PAM aligns each F_i to the spatial resolution and channel dimension of F_u using interpolation and a $1 \times 1 \times 1$ projection. The aligned encoder features are projected into key and value representations, while the decoder feature is projected into a query representation:

$$K_i = W_i^K(F_i), \quad V_i = W_i^V(F_i), \quad Q_u = W_u^Q(F_u), \tag{6}$$

where W_i^K and W_i^V are encoder-side projections for the i-th skip candidate, and W_u^Q is the decoder-side query projection. The path weights are computed from the compatibility between the query Q_u and each encoder key K_i:

$$\alpha_i = \frac{\exp(\langle Q_u, K_i \rangle)}{\sum_{j=1}^{M} \exp(\langle Q_u, K_j \rangle)}, \; Z_u = \sum_{i=1}^{M} \alpha_i V_i. \tag{7}$$

The aggregated skip feature Z_u is residually fused with the current decoder feature as $P_u = F_u + Z_u$, and (F_u, P_u) is fed into the corresponding US-Cell, where α_i denotes the scalar weight of the i-th skip candidate and $\langle Q_u, K_i \rangle$ denotes the scaled mean element-wise product over spatial and channel dimensions. Through soft path weighting, PAM keeps multiple skip candidates available instead of selecting a single discrete path. This enables adaptive multi-scale feature fusion while avoiding enumeration of skip-path combinations.

4 EXPERIMENTS

4.1 BENCHMARKS AND IMPLEMENTATION

Datasets and implementation. We evaluate mCNA on Synapse (Fang & Yan, 2020), a benchmark for comparison with prior abdominal multi-organ segmentation methods, and all 10 tasks of the Medical Segmentation Decathlon (MSD) (Antonelli et al., 2022) to assess cross-task generalization across medical imaging tasks. Synapse contains 30 abdominal CT volumes with eight-organ annotations, using 18 volumes for training and 12 for validation. For MSD, task-specific training and validation splits are constructed from the provided cases, and all baselines follow the same splits, preprocessing, augmentation, and evaluation settings. 2D experiments on CHAOS CT/MR (Kavur et al., 2021), BAGLS (Gómez et al., 2020), and KVASIR (Pogorelov et al., 2017) are provided in Appendix F, with convolution, pooling, interpolation, and LDW operations instantiated in 2D.

For Synapse, all volumes are resampled to a spacing of $1.5 \times 1.5 \times 2.0$ using trilinear interpolation for images and nearest-neighbor interpolation for labels. For each MSD task, all compared methods use the same target voxel spacing, preprocessing procedures, and training/validation split. The task-wise splits are summarized in Appendix D.5. Models are trained using randomly cropped 128^3 and 96^3 patches for Synapse and MSD, respectively, and evaluated through sliding-window inference with equally sized windows and 50% overlap. Dice and HD95 are averaged over foreground classes. All models are implemented in PyTorch with MONAI and trained on a single NVIDIA A100 80GB GPU using a hybrid Dice and cross-entropy loss. We use 200 iterations per epoch, with batch sizes of 1 and 4 for search and retraining, respectively.

Architecture search and retraining. Following the search-once-transfer protocol of DiNTS (He et al., 2021), we use MSD Task07 Pancreas as the proxy search task and split its training volumes into $\mathcal{D}_w$ and $\mathcal{D}_\alpha$ with an 80/20 ratio for bi-level optimization. The search runs for 150 epochs, using AdamW for network weights and Adam for architecture parameters, both with an initial learning rate of 1×10^{-3}. Network weights use a weight decay of 1×10^{-5} and a warmup cosine schedule with 10 warmup epochs and $\eta_{\min} = 1 \times 10^{-6}$. The differentiable DS/US cell search takes 0.5 days on one A100 GPU; clone-number selection is performed separately through post-search validation.

After discretization, the derived 3D DS/US cells are transferred to Synapse and all MSD tasks without task-specific architecture search. Each resulting network is retrained from scratch for 400 epochs using AdamW with the same learning rate, weight decay, and warmup cosine schedule. Mixed-precision training is enabled, and retraining requires nearly 1.5 days on one A100 GPU. We use $N = 3$ for all main experiments; the detailed clone-number sensitivity and accuracy–efficiency analysis are provided in Appendix E.4.

4.2 COMPARISON WITH EXISTING METHODS AND COMPONENT ANALYSIS

We compare mCNA with representative 3D segmentation backbones, including CNN-based encoder–decoder models (3D U-Net (Çiçek et al., 2016), SegResNet (Myronenko, 2018), 3D UX-Net (Lee et al., 2022), MedNeXt (Roy et al., 2023)), Transformer-based architectures (SwinUNETR (Hatamizadeh et al., 2021)), and recent state-space-model-based backbones (SegMamba (Xing et al., 2024), U-Mamba (Ma et al., 2024), SegMamba-V2 (Xing et al., 2025)). In addition, we include DiNTS (He et al., 2021) as a representative 3D NAS method. Compared with 2D NAS, 3D NAS methods remain limited due to the substantial memory and computational costs of volumetric architecture search. Since most existing 3D NAS methods do not release source code, we primarily compare with DiNTS under the same experimental protocol. DiNTS is re-searched and retrained under our aligned settings and evaluated on both Synapse and all ten MSD tasks.

Table 1: Quantitative comparison on the Synapse multi-organ CT dataset under an aligned evaluation protocol. Organ-wise Dice scores (%) and overall Dice and HD95 values are reported as mean ± standard deviation over three independent runs. Best and second-best values are **bolded** and underlined, respectively.

Method	Spleen	Left Kidney	Right Kidney	Gallbladder	Liver
3D U-Net	89.22 ± 1.18	85.79 ± 1.18	84.99 ± 0.18	61.32 ± 3.10	95.39 ± 0.07
SegResNet	91.61 ± 1.53	84.60 ± 1.47	85.51 ± 0.28	63.76 ± 4.89	95.15 ± 0.35
3D UX-Net	90.48 ± 1.24	86.07 ± 0.42	84.27 ± 0.71	63.58 ± 1.91	95.58 ± 0.29
MedNeXt	87.05 ± 1.36	85.67 ± 0.67	84.96 ± 1.39	65.25 ± 1.76	**95.77** ± 0.22
SwinUNETR	89.00 ± 1.73	85.44 ± 0.29	83.96 ± 1.17	64.61 ± 1.89	95.19 ± 0.28
SegMamba	90.10 ± 1.80	86.63 ± 0.84	85.73 ± 0.30	66.76 ± 2.82	94.70 ± 1.02
U-Mamba	91.85 ± 1.13	85.88 ± 0.50	86.17 ± 0.37	62.67 ± 0.86	95.76 ± 0.18
SegMamba-V2	90.92 ± 0.65	86.31 ± 0.77	85.82 ± 1.03	68.45 ± 0.66	95.51 ± 0.63
DiNTS	87.50 ± 1.26	85.41 ± 0.57	84.34 ± 0.66	68.55 ± 3.01	93.97 ± 0.37
mCNA + Orth. LDW	**93.41** ± 0.20	**86.71** ± 0.07	**86.19** ± 0.12	**76.07** ± 0.97	95.34 ± 0.06

Method	Stomach	Aorta	Pancreas	Dice (↑)	HD95 (↓)
3D U-Net	78.44 ± 0.54	90.41 ± 0.33	65.19 ± 0.52	81.35 ± 0.59	21.95 ± 2.83
SegResNet	76.09 ± 0.67	90.49 ± 1.22	65.09 ± 3.34	81.54 ± 0.09	33.92 ± 3.23
3D UX-Net	75.19 ± 1.26	89.24 ± 0.57	64.18 ± 0.97	81.07 ± 0.17	34.41 ± 2.08
MedNeXt	78.18 ± 1.74	89.78 ± 0.68	70.07 ± 2.09	82.09 ± 0.20	32.64 ± 6.05
SwinUNETR	75.37 ± 0.88	89.81 ± 0.77	65.46 ± 0.74	81.11 ± 0.18	36.61 ± 7.09
SegMamba	77.13 ± 0.89	91.22 ± 0.22	69.51 ± 0.77	82.72 ± 0.50	30.69 ± 8.11
U-Mamba	78.92 ± 2.81	91.69 ± 0.23	68.46 ± 1.56	82.68 ± 0.15	31.50 ± 2.32
SegMamba-V2	77.52 ± 1.32	91.68 ± 0.28	71.15 ± 2.19	83.42 ± 0.44	28.34 ± 9.56
DiNTS	75.97 ± 1.07	89.68 ± 0.44	65.32 ± 1.15	81.34 ± 0.56	23.76 ± 4.33
mCNA + Orth. LDW	**80.44** ± 0.39	**91.72** ± 0.15	**71.23** ± 0.75	**85.14** ± 0.16	**9.23** ± 1.68

Results on Synapse. Table 1 reports organ-wise Dice, mean Dice, and HD95 on Synapse under the aligned protocol, with results reported as mean ± standard deviation over three independent runs. mCNA + Orth. LDW achieves the highest mean Dice of 85.14 ± 0.16 and the lowest HD95 of 9.23 ± 1.68 among compared methods. Compared with SegMamba-V2, the baseline with the highest mean Dice, our model improves mean Dice by 1.72 percentage points and reduces HD95 by 19.11. Compared with DiNTS, it improves mean Dice by 3.80 percentage points and reduces HD95 by 14.53. Organ-wise, our model achieves the highest mean Dice on seven of the eight organs, except the liver. The largest improvement over the strongest baseline for each organ occurs on the gallbladder, where Dice increases from 68.55 to 76.07, a gain of 7.52 percentage points. The wavelet-based downsampling ablation comparing mCNA, mCNA + Haar, and mCNA + Orth. LDW is provided in Appendix E.1. The complementary effects of the searched cells, CU, and PAM are analyzed in Table 3, while qualitative results on Synapse and MSD are provided in Appendix G.

Results on the complete MSD benchmark. Table 2 reports Dice scores on the complete 10-task MSD benchmark under the aligned validation protocol. The MSD results are computed on our task-wise validation splits rather than the official hidden test set, and all baselines are retrained with the same splits and preprocessing. We compare mCNA + Orth. LDW with MedNeXt, SwinUNETR, SegMamba, U-Mamba, SegMamba-V2, and DiNTS across all MSD tasks.

When averaging the task-level Dice scores equally across all ten tasks, mCNA + Orth. LDW achieves the highest 10-task macro-average Dice of 70.92, compared with 68.39 for DiNTS. At the individual-task level, mCNA + Orth. LDW achieves the highest mean Dice on nine of the ten tasks. The exception is Task05 Prostate, where DiNTS achieves 71.80 Dice compared with 63.93 for mCNA + Orth. LDW.

Among the tasks where our method performs best, the largest improvements over the strongest baseline are observed on Lung and Colon, with gains of 9.56 and 7.83 percentage points, respec-

Table 2: Mean Dice scores (%) on the ten MSD tasks under aligned protocols. Multi-class scores are averaged across target classes, and the 10-task average is the task-level macro-average. Best and second-best values are **bolded** and underlined, respectively. Class-wise results are provided in Appendix E.2.

Method	Brain	Heart	Liver	Hippo.	Prostate	Lung	Pancreas	Hepatic V.	Spleen	Colon	10-Task Avg.
MedNeXt	78.37	91.85	62.70	87.25	52.73	32.21	61.29	59.46	85.76	22.07	63.37
SwinUNETR	78.22	90.72	54.84	87.07	53.21	16.69	58.07	55.96	79.51	31.24	60.55
SegMamba	77.92	92.47	57.62	87.51	57.00	34.17	58.41	62.00	77.14	20.82	62.51
U-Mamba	77.20	83.22	58.43	87.06	57.83	37.60	59.17	62.39	86.26	22.62	63.18
SegMamba-V2	77.85	90.22	62.12	87.26	60.52	44.51	60.34	61.35	91.46	24.42	66.01
DiNTS	74.36	92.46	63.57	87.69	**71.80**	47.97	59.59	61.61	93.98	30.88	68.39
mCNA + Orth. LDW	**79.25**	**92.73**	**68.68**	**87.77**	63.93	**57.53**	**61.85**	**62.86**	**95.53**	**39.07**	**70.92**

Table 3: Incremental component ablation on Synapse with $N = 3$. Results are from a single run; full configurations are provided in Appendix E.3

NAS	CU	PAM	mDice (↑)	HD95 (↓)	Params (M)
✗	✗	✗	81.89	17.23	27.71
✓	✗	✗	82.58	15.54	26.54
✓	✓	✗	83.25	15.39	42.47
✓	✓	✓	**85.36**	**7.54**	42.86

tively. Overall, these results demonstrate that the searched multi-clone communication architecture transfers effectively across diverse MSD tasks without task-specific architecture search. Complete class-wise results are provided in Appendix E.2.

Component analysis. Table 3 presents an incremental analysis of the main components. Starting from the hand-crafted baseline without CU or PAM, replacing the fixed cells with the searched DS/US cells improves mDice from 81.89 to 82.58 and reduces HD95 from 17.23 to 15.54. Adding CU further improves mDice to 83.25 and slightly reduces HD95 to 15.39. Incorporating PAM raises mDice to 85.36 and reduces HD95 to 7.54, yielding the best values of both metrics among the evaluated configurations. Additional component, communication, wavelet, and clone-number ablations are provided in Appendices E.3, E.7, E.1, and E.4, respectively.

5 Conclusion

We propose mCNA, a search-efficient NAS framework for 3D medical image segmentation that decouples cell-level search from macro-level construction. Combining clone-based communication, PAM-based skip fusion, and wavelet downsampling, mCNA achieves the highest mean Dice and lowest HD95 on Synapse and the highest task-level Dice on nine of the ten MSD tasks among the evaluated methods under aligned experimental protocols. Ablation studies validate the complementary contributions of these components. The searched architecture transfers across datasets without additional search, while 2D experiments further demonstrate its applicability beyond volumetric segmentation.

AI USE STATEMENT

We used ChatGPT to assist with manuscript drafting and language revision, checking consistency between reported numerical results and their descriptions, and refining the interpretation of experimental results. The authors reviewed and verified the AI-assisted content against the experimental records and take responsibility for the final manuscript, including its claims, results, and references.

ETHICS STATEMENT

This work investigates medical image segmentation using established benchmark datasets. The reported results assess algorithmic performance and do not establish clinical safety or effectiveness. Segmentation errors and differences across patient populations, institutions, and imaging protocols may affect practical use. Clinical deployment would require further external validation and appropriate clinical oversight.

REPRODUCIBILITY STATEMENT

The proposed architecture and its components are described in Section 3, and the training and evaluation protocols are summarized in Section 4.1. Additional details on the search procedure, primitive operations, wavelet formulation, and experimental settings are provided in Appendices A, B.1, C, and D.

REFERENCES

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Emil Benedykciuk, Marcin Denkowski, and Grzegorz Wójcik. Implantable adaptive cells in u-net skip connections: Low-compute retrofit for medical segmentation. *IEEE Access*, 2026.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation*, pp. 103–111, 2014.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pp. 424–432. Springer, 2016.

Badhan Kumar Das, Ajay Singh, Saahil Islam, Gengyan Zhao, and Andreas Maier. Segres-mamba: An efficient architecture for 3d medical image segmentation. In Tolga Tasdizen, Shireen Elhabian, Ronald Summers, Chen Chen, Lisa Koch, and Yan Zhuang (eds.), *Proceedings of The 8th International Conference on Medical Imaging with Deep Learning*, volume 301 of *Proceedings of Machine Learning Research*, pp. 280–292. PMLR, 09–11 Jul 2026. URL https://proceedings.mlr.press/v301/das26a.html.

Bo Dong, Wenhai Wang, Deng-Ping Fan, Jinpeng Li, Huazhu Fu, and Ling Shao. Polyp-pvt: Polyp segmentation with pyramid vision transformers. *arXiv preprint arXiv:2108.06932*.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 263–273. Springer, 2020a.

Tongle Fan, Guanglei Wang, Yan Li, and Hongrui Wang. Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access*, 8:179656–179665, 2020b.

Xi Fang and Pingkun Yan. Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction. *IEEE Transactions on Medical Imaging*, 39(11):3619–3629, 2020.

D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf. Residual conv-deconv grid network for semantic segmentation. In *The British Machine Vision Conference (BMVC)*, September 2017.

Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet convolutional neural networks. *arXiv preprint arXiv:1805.08620*, 2018.

Pablo Gómez, Andreas M. Kist, Patrick Schlegel, David A. Berry, Dinesh K. Chhetri, Stephan Dürr, Matthias Echternach, Aaron M. Johnson, Stefan Kniesburges, Melda Kunduk, Youri Maryn, Anne Schützenberger, Monique Verguts, and Michael Döllinger. Bagls, a multihospital benchmark for automatic glottis segmentation. *Scientific Data*, 7:1–12, 2020.

Jiayao Hao, Lingling Li, Licheng Jiao, Feiyu Shi, Xu Liu, Ruiyang Li, Zhongjian Huang, Qigong Sun, and Huiyu Zhou. Wavelet spectral-aware kolmogorov-arnold network for organ and tumor segmentation. *Computerized Medical Imaging and Graphics*, pp. 102784, 2026.

Tahereh Hassanzadeh, Daryl Essam, and Ruhul Sarker. 2d to 3d evolutionary deep convolutional neural networks for medical image segmentation. *IEEE Transactions on Medical Imaging*, 40(2): 712–721, 2020.

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pp. 272–284. Springer, 2021.

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 574–584, 2022.

Yufan He, Dong Yang, Holger Roth, Can Zhao, and Daguang Xu. Dints: Differentiable neural network topology search for 3d medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5841–5850, 2021.

Ziyan Huang, Zehua Wang, Zhikai Yang, and Lixu Gu. Adwu-net: adaptive depth and width u-net for medical image segmentation by differentiable neural architecture search. In *International Conference on Medical Imaging with Deep Learning*, pp. 576–589. PMLR, 2022.

A. Emre Kavur, N. Sinem Gezer, Mustafa Barış, Sinem Aslan, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, Bora Baydar, Dmitry Lachinov, Shuo Han, Josef Pauli, Fabian Isensee, Matthias Perkonigg, Rachana Sathish, Ronnie Rajan, Debdoot Sheet, Gurbandurdy Dovletov, Oliver Speck, Andreas Nürnberger, Klaus H. Maier-Hein, Gözde Bozdağı Akar, Gözde Ünal, Oğuz Dicle, and M. Alper Selver. Chaos challenge – combined (ct-mr) healthy abdominal organ segmentation. *Medical Image Analysis*, pp. 229–238, 2021.

Ho Hin Lee, Shunxing Bao, Yuankai Huo, and Bennett A Landman. 3d ux-net: A large kernel volumetric convnet modernizing hierarchical transformer for medical image segmentation. *arXiv preprint arXiv:2209.15076*, 2022.

Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L. Yuille, and Li Fei-Fei. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2019.

Guoqi Liu, Zongyu Chen, Dong Liu, Baofang Chang, and Zhi Dou. Ftmf-net: A fourier transform-multiscale feature fusion network for segmentation of small polyp objects. *IEEE Transactions on Instrumentation and Measurement*, 72:1–15, 2023.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI brainlesion workshop*, pp. 311–320. Springer, 2018.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

Cheng Peng, Andriy Myronenko, Ali Hatamizadeh, Vishwesh Nath, Md Mahfuzur Rahman Siddiquee, Yufan He, Daguang Xu, Rama Chellappa, and Dong Yang. Hypersegnas: Bridging one-shot neural architecture search with 3d medical image segmentation using hypernet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20741–20751, 2022.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 164–169, 2017. ISBN 978-1-4503-5002-0.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351, pp. 234–241, 2015.

Saikat Roy, Gregor Koehler, Constantin Ulrich, Michael Baumgartner, Jens Petersen, Fabian Isensee, Paul F Jaeger, and Klaus H Maier-Hein. Mednext: transformer-driven scaling of convnets for medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 405–415. Springer, 2023.

Bor-Shiun Wang, Jun-Wei Hsieh, Ming-Ching Chang, Ping-Yang Chen, Lipeng Ke, and Siwei Lyu. Learnable discrete wavelet pooling (LDW-Pooling) for convolutional networks. In *The British Machine Vision Conference (BMVC)*, 2021a.

Haonan Wang, Peng Cao, Jinzhu Yang, and Osmar Zaiane. Narrowing the semantic gaps in u-net with learnable skip connections: The case of medical image segmentation. *Neural Networks*, 178: 106546, 2024.

Jiacheng Wang, Lan Wei, Liansheng Wang, Qichao Zhou, Lei Zhu, and Jing Qin. Boundary-aware transformers for skin lesion segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, pp. 206–216. Springer, 2021b.

Jiacheng Wang, Fei Chen, Yuxi Ma, Liansheng Wang, Zhaodong Fei, Jianwei Shuai, Xiangdong Tang, Qichao Zhou, and Jing Qin. Xbound-former: Toward cross-scale boundary modeling in transformers. *IEEE Transactions on Medical Imaging*, 42(6):1735–1745, 2023.

Xinyi Wang, Tiange Xiang, Chaoyi Zhang, Yang Song, Dongnan Liu, Heng Huang, and Weidong Cai. Bix-nas: Searching efficient bi-directional architecture for medical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 229–238, 2021c.

Jiahong Wei, Bing Xue, and Mengjie Zhang. Multi-objective once-for-all neural architecture search for medical image segmentation. *IEEE Transactions on Image Processing*, 2026.

Ruxue Wen, Hangjie Yuan, Dong Ni, Wenbo Xiao, and Yaoyao Wu. From denoising training to test-time adaptation: Enhancing domain generalization for medical image segmentation. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 453–463, 2024.

Y. Weng, T. Zhou, Y. Li, and X. Qiu. NAS-Unet: Neural architecture search for medical image segmentation. *IEEE Access*, 7:44247–44257, 2019.

Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. In *International conference on learning representations*, 2018.

Bin Xie, Yan Yan, and Gady Agam. MM-UNet: Meta Mamba UNet for Volumetric Medical Image Segmentation . In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2026*, volume LNCS 16883. Springer Nature Switzerland, September 2026.

Z Xing, T Ye, Y Yang, G Liu, and L Zhu. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. arxiv 2024. *arXiv preprint arXiv:2401.13560*, 2024.

Zhaohu Xing, Tian Ye, Yijun Yang, Du Cai, Baowen Gai, Xiao-Jian Wu, Feng Gao, and Lei Zhu. Segmamba-v2: Long-range sequential modeling mamba for general 3d medical image segmentation. *IEEE Transactions on Medical Imaging*, 2025.

Shuying Xu and Hongyan Quan. Ect-nas: Searching efficient cnn-transformers architecture for medical image segmentation. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1601–1604, 2021. doi: 10.1109/BIBM52615.2021.9669734.

Xingang Yan, Weiwen Jiang, Yiyu Shi, and Cheng Zhuo. Ms-nas: Multi-scale neural architecture search for medical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 388–397, 2020.

Caiyang Yu, Yixi Wang, Chenwei Tang, Wentao Feng, and Jiancheng Lv. Eu-net: Automatic u-net neural architecture search with differential evolutionary algorithm for medical imagesegmentation. *Computers in Biology and Medicine*, 167, 2023.

Qihang Yu, Dong Yang, Holger Roth, Yutong Bai, Yixiao Zhang, Alan L. Yuille, and Daguang Xu. C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4125–4134, 2020.

Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. *arXiv preprint arXiv:1909.09656*, 2019.

Haokui Zhang, Ying Li, Hao Chen, and Chunhua Shen. Memory-efficient hierarchical neural architecture search for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3657–3666, 2020a.

Ruifei Zhang, Guanbin Li, Zhen Li, Shuguang Cui, Dahong Qian, and Yizhou Yu. Adaptive context selection for polyp segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23*, pp. 253–262. Springer, 2020b.

Yundong Zhang, Huiye Liu, and Qiang Hu. Transfuse: Fusing transformers and cnns for medical image segmentation. In *Medical image computing and computer assisted intervention–MICCAI 2021: 24th international conference, Strasbourg, France, September 27–October 1, 2021, proceedings, Part I 24*, pp. 14–24. Springer, 2021.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

A Search-and-Expansion Algorithm

The complete search-and-expansion procedure of mCNA is summarized in Algorithm 1. The framework first searches the DS and US cell architectures within a PAM-enabled 3D U-shaped supernet through bi-level optimization. The searched cells are then discretized by retaining the two strongest incoming edges for each intermediate node and selecting the highest-weighted operation on each retained edge. Finally, the resulting single-clone backbone is expanded to different clone numbers, where adjacent clones communicate through CUs. The clone number is selected through post-search validation. The reported search-stage cost corresponds to cell-level supernet optimization, while clone-configuration training and validation belong to the subsequent model-selection stage.

Algorithm 1 Search-and-expansion procedure of mCNA.

Require: Proxy dataset $\mathcal{D}_p$; target sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$; candidate clone numbers $\mathcal{N}$; search epochs E_s; warm-up epochs $E_w = 50$; architecture-update interval $q = 2$; perturbation magnitude $\epsilon_\alpha = 0.3$

Ensure: Final architecture $\mathbf{A}^\star$

Split $\mathcal{D}_p$ into $\mathcal{D}_w$ and $\mathcal{D}_\alpha$ for network-weight and architecture-parameter optimization.
Construct a PAM-enabled 3D U-shaped supernet $\mathcal{S}(w, \alpha)$, where each searchable edge is a softmax-weighted mixture of candidate operations.
for $e = 1$ to E_s **do**
  Update network weights w on $\mathcal{D}_w$ while fixing α.
  if $e > E_w$ and $e \bmod q = 0$ **then**
    Sample random noise ξ and compute $\tilde{\alpha} = \alpha + \epsilon_\alpha \xi$.
    Update architecture parameters α on $\mathcal{D}_\alpha$ using the perturbed weights $\tilde{\alpha}$ while fixing w.
  end if
end for
for each intermediate node j in each DS/US cell **do**
  Score each incoming edge (i, j) by $s_{i,j} = \max_{o \in \mathcal{O}} \alpha_o^{(i,j)}$.
  Retain the two incoming edges with the largest $s_{i,j}$ values.
  On each retained edge, select $o_{i,j}^\star = \arg\max_{o \in \mathcal{O}} \alpha_o^{(i,j)}$.
end for
Assemble the discretized DS/US cells into the reusable single-clone backbone $\mathbf{A}$.
for $N \in \mathcal{N}$ **do**
  Construct $\mathbf{A}_N$ by cloning $\mathbf{A}$ N times.
  Equip each clone with a CU at its bottleneck and apply PAM for decoder skip aggregation.
  Initialize the incoming message as $m_0 = 0$.
  for $n = 1$ to N **do**
    Obtain the updated communication state and decoder injection: $(m_n, d_n) = \mathrm{CU}(m_{n-1}, e_n)$.
    Inject d_n into the decoder and propagate m_n to the next clone.
  end for
  Fuse $\{\widehat{Y}_1, \ldots, \widehat{Y}_N\}$ using the final refinement head.
  Train $\mathbf{A}_N$ on $\mathcal{D}_{\text{train}}$ and evaluate it on $\mathcal{D}_{\text{val}}$.
end for
Select $N^\star = \arg\max_{N \in \mathcal{N}} \mathrm{Dice}(\mathbf{A}_N; \mathcal{D}_{\text{val}})$, using computational efficiency as a secondary criterion for configurations with comparable Dice.
Set $\mathbf{A}^\star = \mathbf{A}_{N^\star}$.
return $\mathbf{A}^\star$.

B Search Space and Optimization Details

B.1 Primitive Operation Sets

This section provides the complete primitive operation set used in the cell-level 3D cell architecture search. The final searched cell architectures under different meta-node settings are visualized in Fig. 4.

We use a DARTS-style differentiable architecture search strategy (Liu et al., 2018) to optimize the cell-level 3D U-shaped supernet. Each searchable edge is represented as a softmax-weighted mixture of candidate operations during search, and the final discrete architecture is obtained by selecting the strongest incoming edges and the highest-weighted operation on each selected edge.

Following the bi-level optimization setting, we alternately update network weights and architecture parameters on two disjoint subsets of MSD Task07 Pancreas. Architecture-parameter updates start after a warm-up period of 50 epochs. After the warm-up stage, architecture parameters are updated every 2 epochs, while network weights are updated throughout the search stage. We apply gradient clipping with a maximum norm of 5 to stabilize optimization.

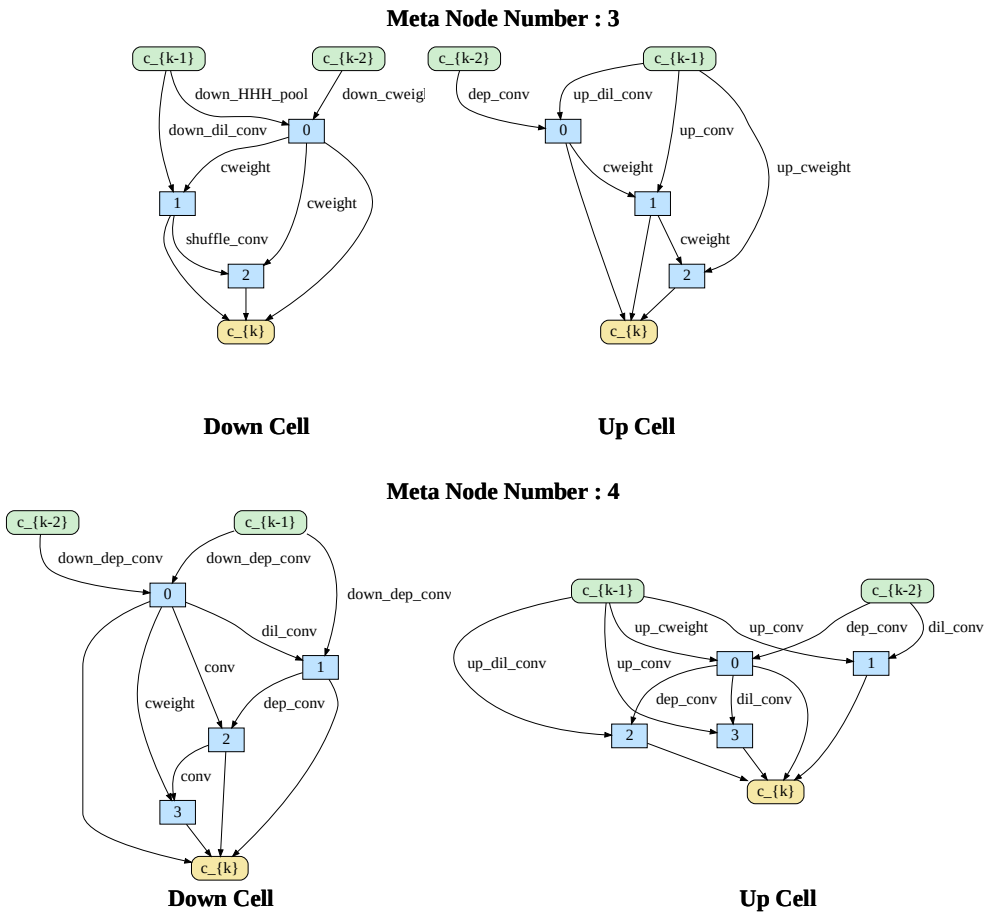

Figure 4: Comparison of cell architectures searched on MSD Task07 Pancreas with different meta-node numbers. The upper row shows the default searched downsampling and upsampling cells with 3 meta-nodes used in the main experiments, while the lower row shows the corresponding cells with 4 meta-nodes.

To reduce the sensitivity of the final discretized architecture to small variations in architecture weights, we additionally apply random perturbation to the architecture parameters during search. Specifically, before discretization-sensitive architecture updates, we perturb the architecture parameters as

$$\tilde{\alpha}^{(i,j)} = \alpha^{(i,j)} + \epsilon_\alpha \cdot \xi^{(i,j)}, \tag{8}$$

where $\xi^{(i,j)}$ is sampled from a random noise distribution with the same shape as $\alpha^{(i,j)}$, and $\epsilon_\alpha = 0.3$ controls the perturbation magnitude. This perturbation encourages the searched architecture to remain stable under small changes in architecture weights, which is motivated by prior studies showing that DARTS-style search can be sensitive to architecture-weight perturbations and discretization effects (Zela et al., 2019). In our implementation, we use random architecture-parameter perturbation, denoted as `perturb_alpha=random`, with `epsilon_alpha=0.3`.

We use 3 meta-nodes as the default cell configuration. The search-stage comparison with DiNTS and the effect of using four meta-nodes are presented in the section "Search-stage Comparison and Meta-node Ablation."

Table 4: Primitive operation sets for 3D cell architecture search. The downsampling operation set includes conventional pooling, learnable convolutional downsampling, and LDW-based structure-preserving pooling candidates.

Normal Ops
channel-weighted conv
dilated conv
depthwise separable conv
channel shuffle conv
conv
identity

Up Ops
Up channel weighted conv
Up depthwise separable conv
Up conv
Up dilated conv

Down Ops	
average pooling	max pooling
Down depthwise separable conv	Down conv
Down channel-weighted conv	LLL pooling
Down dilated conv	HLL pooling
LHL pooling	LLH pooling
HHL pooling	HLH pooling
LHH pooling	HHH pooling

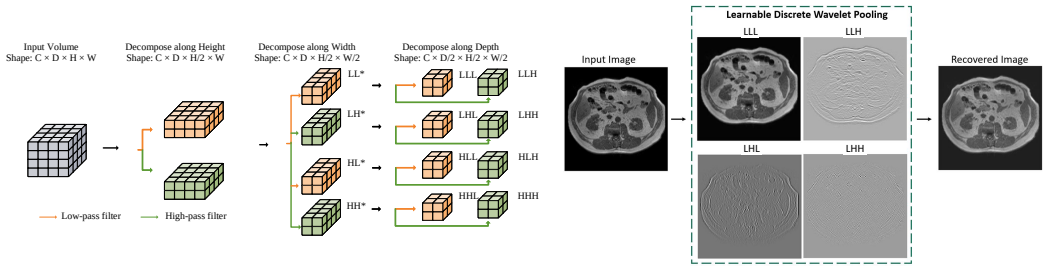

Figure 5: Illustration of the proposed volumetric LDW pooling. Left: a 3D feature volume is decomposed separably along the height, width, and depth axes, producing eight volumetric subbands: LLL, LLH, LHL, LHH, HLL, HLH, HHL, and HHH. Orange and green arrows indicate low-pass and high-pass filtering, respectively. Right: representative slices extracted from the input volume, selected 3D LDW subbands, and the reconstructed volume are shown for visualization.

C Volumetric LDW Formulation and Regularization Details

As shown in Fig. 5, the proposed volumetric LDW pooling performs separable low-/high-pass decomposition along three spatial axes. Given a 3D feature volume $X \in \mathbb{R}^{C\times D\times H\times W}$, the decomposition is applied along the height, width, and depth axes. Each axis produces one low-pass and one high-pass response, and their combinations form eight volumetric subbands:

$$\{LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH\}.$$

Each letter denotes whether a low-pass (L) or high-pass (H) filter is applied along one spatial axis. In our implementation, these subbands are used as candidate downsampling operations in the 3D cell search space. For visualization, the right side of Fig. 5 shows representative 2D slices extracted from the corresponding 3D subbands rather than independent 2D LDW outputs.

To stabilize the learnable low-pass and high-pass filters used in volumetric LDW pooling, we follow the wavelet regularization constraints below. The square sum of filter weights for the low-pass DW filter should equal 1. We adopt the following loss term described in (Wang et al., 2021a) for the

low-pass DW filters:

$$L_{Low} = \left[\sum_{i\in\mathbb{K}} W^L(i)^2 - 1\right]^2 + \left[\sum_{i\in\mathbb{K}} W^L(i) - \sqrt{2}\right]^2. \tag{9}$$

For the high-pass DW filters, since their sum of weights must be zero, we adopt the following constraint:

$$L_{High} = \left(\sum_{i\in\mathbb{K}} W^H(i)^2 - 1\right)^2 + \left(\sum_{i\in\mathbb{K}} W^H(i)\right)^2, \tag{10}$$

where the first term of equation 10 enforces W^H to become unit vectors and the second term enforces the sum of coefficients in the high-pass filter W^H to be zero.

To retain the *reversibility* of DW pooling, the energy of W^L and W^H should equal 1. Thus, we introduce the following reversibility loss:

$$L_{Reverse} = \left[\left(\sum_{i\in\mathbb{K}} W^L(i)^2 + \sum_{i\in\mathbb{K}} W^H(i)^2\right) - 2\right]^2. \tag{11}$$

The three constraints in equation 9–equation 11 jointly ensure: (1) the low-pass filter weights sum to $\sqrt{2}$, (2) the high-pass filter weights sum to zero, and (3) the pooling operation remains reversible. However, the high-pass filter and the low-pass filter should also be orthogonal to each other. Different from (Wang et al., 2021a), a new orthogonality constraint is added in this paper:

$$L_{Orthogonal} = \left(\sum_{i\in\mathbb{K}} W^L(i)W^H(i)\right)^2. \tag{12}$$

Based on equation 9–equation 12, the four terms are summed up for training the filters W^L and W^H:

$$L_{Wavelet} = L_{Low} + L_{High} + L_{Reverse} + L_{Orthogonal}. \tag{13}$$

D Additional Experimental Settings

We provide additional experimental details and analyses that complement the main paper.

Unless otherwise specified, all experiments reported in this appendix follow the same preprocessing, training protocol, and evaluation metrics as described in the main paper. For architecture-related comparisons, we report both accuracy and efficiency to highlight the trade-off between segmentation performance and computational cost. For search-related comparisons, we distinguish between search-stage and retraining-stage differences to clarify whether performance gains arise from improved architecture discovery or increased model capacity. The main Synapse comparisons and the wavelet-based downsampling ablation are reported as the mean and standard deviation over three independent runs. MSD results and all remaining ablation results are obtained from a single training run.

D.1 Hyperparameter Selection

Key architectural hyperparameters were selected based on validation Dice and computational efficiency. We evaluated U-Net depths $\{4, 5\}$, meta-node numbers $\{3, 4\}$, and clone numbers $N \in \{1, 2, 3, 4\}$, adopting the setting with the best validation Dice–efficiency trade-off.

D.2 Random Seed and Data Splits

A fixed random seed of 42 was used to construct the training and validation splits, and the same splits were used for all compared methods.

D.3 Computing Infrastructure

All architecture search, retraining, and evaluation experiments were conducted on a single NVIDIA A100 80GB PCIe GPU with 80 GB of GPU memory. The compute node was equipped with an Intel Xeon Silver 4410Y CPU and 251 GiB of system memory, running Ubuntu 22.04.5 LTS. Our implementation used Python 3.11.13, PyTorch 2.8.0, MONAI 1.5.2, CUDA 12.9, and cuDNN 9.10.2. Unless otherwise specified, all 3D comparison methods retrained under our aligned experimental protocol were executed using the same hardware and software environment.

D.4 Evaluation Metrics

Let Ω denote the pixel or voxel domain. For the n-th case and foreground class c, let

$$\begin{aligned} P_{n,c} &= \{x \in \Omega : \hat{y}_n(x) = c\}, \\ G_{n,c} &= \{x \in \Omega : y_n(x) = c\}. \end{aligned} \tag{14}$$

where $P_{n,c}$ and $G_{n,c}$ denote the predicted and ground-truth regions, respectively. Background is excluded from the reported class-averaged metrics unless otherwise specified.

Dice similarity coefficient. The Dice score for case n and class c is defined as

$$\mathrm{Dice}_{n,c} = \frac{2|P_{n,c} \cap G_{n,c}|}{|P_{n,c}| + |G_{n,c}|}. \tag{15}$$

Following our implementation, when both $P_{n,c}$ and $G_{n,c}$ are empty, $\mathrm{Dice}_{n,c}$ is defined as 1; when exactly one of them is empty, it is defined as 0. The case-level mean Dice is

$$\mathrm{mDice}_n = \frac{1}{C} \sum_{c=1}^{C} \mathrm{Dice}_{n,c}, \tag{16}$$

and the overall reported mean Dice is

$$\mathrm{mDice} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{mDice}_n, \tag{17}$$

where C and N denote the numbers of foreground classes and evaluation cases, respectively.

Surface distance. Let $\partial P_{n,c}$ and $\partial G_{n,c}$ denote the surfaces or boundaries of the predicted and ground-truth regions. Given the pixel or voxel spacing vector $\mathbf{s} = (s_1, \ldots, s_K)$, the spacing-aware Euclidean distance between two boundary points x and y is

$$d_{\mathbf{s}}(x, y) = \sqrt{\sum_{k=1}^{K} s_k^2 (x_k - y_k)^2}. \tag{18}$$

The distance from a point x to a boundary set S is

$$d_{\mathbf{s}}(x, S) = \min_{y \in S} d_{\mathbf{s}}(x, y). \tag{19}$$

The two directed surface-distance sets are

$$\mathcal{D}_{P \to G} = \{d_{\mathbf{s}}(p, \partial G_{n,c}) : p \in \partial P_{n,c}\}, \tag{20}$$

$$\mathcal{D}_{G \to P} = \{d_{\mathbf{s}}(g, \partial P_{n,c}) : g \in \partial G_{n,c}\}. \tag{21}$$

Hausdorff distance. The symmetric Hausdorff distance is defined as

$$\mathrm{HD}_{n,c} = \max\{\max \mathcal{D}_{P \to G}, \max \mathcal{D}_{G \to P}\}. \tag{22}$$

For the 2D experiments, distances are computed in pixel units.

Table 5: Task-wise training and validation splits for the MSD benchmark. All compared methods use the same splits and patch sampling protocol.

Task	Training cases	Validation cases	Samples per volume
Task01 BrainTumour	388	96	8
Task02 Heart	16	4	8
Task03 Liver	105	26	8
Task04 Hippocampus	208	52	8
Task05 Prostate	26	6	8
Task06 Lung	51	12	8
Task07 Pancreas	225	56	8
Task08 Hepatic Vessel	243	60	8
Task09 Spleen	33	8	8
Task10 Colon	101	25	8

95th-percentile Hausdorff distance. For the 3D experiments, we report the symmetric 95th-percentile Hausdorff distance:

$$\mathrm{HD95}_{n,c} = \max\left\{Q_{0.95}(\mathcal{D}_{P\to G}), Q_{0.95}(\mathcal{D}_{G\to P})\right\}, \tag{23}$$

where $Q_{0.95}(\cdot)$ denotes the 95th percentile. The distance is computed using the task-specific voxel spacing and is therefore reported in physical units. In our implementation, HD95 is set to 0 when both masks are empty. When exactly one mask is empty, the value is treated as undefined and excluded from HD95 averaging.

Boundary F1 score. For the 2D experiments, let τ denote the boundary-matching distance tolerance. Boundary precision and boundary recall are defined as

$$\mathrm{BP}_{n,c} = \frac{|\{p \in \partial P_{n,c} : d(p, \partial G_{n,c}) \le \tau\}|}{|\partial P_{n,c}|}, \tag{24}$$

$$\mathrm{BR}_{n,c} = \frac{|\{g \in \partial G_{n,c} : d(g, \partial P_{n,c}) \le \tau\}|}{|\partial G_{n,c}|}. \tag{25}$$

The Boundary F1 Score is their harmonic mean:

$$\mathrm{BF1}_{n,c} = \frac{2\,\mathrm{BP}_{n,c}\mathrm{BR}_{n,c}}{\mathrm{BP}_{n,c} + \mathrm{BR}_{n,c}}. \tag{26}$$

Dice measures region-level overlap, whereas HD, HD95, and BF1 measure boundary agreement. HD95 is used for the main 3D experiments because it captures substantial surface deviations while being less sensitive than the maximum HD to isolated outlier points. Together, these region- and boundary-based metrics provide a more comprehensive evaluation of segmentation quality.

D.5 Task-wise MSD Preprocessing and Validation Protocol

The Medical Segmentation Decathlon (MSD) benchmark provides training cases with labels and hidden test cases without public ground-truth annotations. In this work, we report results on validation splits constructed from the available labeled training cases rather than on the official hidden test set. We attempted to submit predictions to the official MSD online evaluation system, but no evaluation scores were returned. We also observed that the public leaderboard had not shown new updates after March, and our inquiry to the organizers in May had not received a response by the time of submission. Therefore, we use fixed task-wise validation splits and retrain all compared methods under the same preprocessing and evaluation protocol to provide a controlled and reproducible comparison.

For each MSD task, all methods use the same target spacing, training/validation split, patch size, patch sampling strategy, and sliding-window inference setting. Images are resampled to the task-specific voxel spacing using trilinear interpolation, while labels are resampled using nearest-neighbor interpolation. During training, we randomly sample eight 96^3 patches from each training

Table 6: Ablation of wavelet-based downsampling on the Synapse dataset. Organ-wise Dice scores (%), overall Dice, and HD95 are reported as mean ± standard deviation over three independent runs. The best mean value among the three variants is highlighted in **bold**.

Method	Spleen	Left Kidney	Right Kidney
mCNA	92.13 ± 0.20	86.30 ± 0.11	85.91 ± 0.20
mCNA + Haar	91.52 ± 0.64	86.05 ± 0.16	84.87 ± 0.40
mCNA + Orth. LDW	**93.41** ± 0.20	**86.71** ± 0.07	**86.19** ± 0.12

Method	Gallbladder	Liver	Stomach
mCNA	74.14 ± 1.59	**96.07** ± 0.16	78.02 ± 0.84
mCNA + Haar	75.09 ± 0.87	94.97 ± 0.06	**81.81** ± 0.42
mCNA + Orth. LDW	**76.07** ± 0.97	95.34 ± 0.06	80.44 ± 0.39

Method	Aorta	Pancreas	Dice (↑)	HD95 (↓)
mCNA	**91.73** ± 0.24	69.52 ± 1.13	84.22 ± 0.03	10.45 ± 0.99
mCNA + Haar	90.01 ± 0.37	65.74 ± 1.15	83.76 ± 0.12	13.77 ± 0.90
mCNA + Orth. LDW	91.72 ± 0.15	**71.23** ± 0.75	**85.14** ± 0.16	**9.23** ± 1.68

volume. During inference, we use sliding-window prediction with a 50% overlap. The task-wise training and validation counts and the number of sampled patches per volume are summarized in Table 5.

E Additional Ablation Studies

E.1 Ablation of Wavelet-Based Downsampling

We compare three variants of mCNA to evaluate the effect of incorporating wavelet-based downsampling into the search space. For mCNA, the search space does not include any wavelet-based downsampling operation. For mCNA + Haar, fixed Haar wavelet pooling is included as a candidate downsampling operation in the search space. For mCNA + Orth. LDW, subband operations generated using LDW filters learned with normalization, reversibility, and orthogonality constraints are included as candidate downsampling operations in the search space. All three variants use the same clone number, communication mechanism, PAM-based skip aggregation, search and retraining protocols, and evaluation setting, while differing in the wavelet-based downsampling operations included in their search spaces.

As shown in Table 6, fixed Haar pooling reduces mean Dice from 84.22 ± 0.03 to 83.76 ± 0.12 and increases HD95 from 10.45 ± 0.99 to 13.77 ± 0.90 relative to mCNA. Thus, fixed Haar pooling does not improve either aggregate metric in this setting. In comparison, mCNA + Orth. LDW achieves 85.14 ± 0.16 mean Dice and 9.23 ± 1.68 HD95, improving Dice by 0.92 percentage points and reducing HD95 by 1.22 relative to mCNA.

At the organ level, mCNA + Orth. LDW achieves the highest mean Dice on five of the eight organs: spleen, left kidney, right kidney, gallbladder, and pancreas. Its largest gains over mCNA occur on the stomach, gallbladder, and pancreas, with improvements of 2.42, 1.93, and 1.71 percentage points, respectively. However, fixed Haar pooling achieves the highest stomach Dice, whereas mCNA achieves the highest liver and aorta Dice. These results show that Orth. LDW provides the highest mean Dice and lowest HD95 among the evaluated variants on Synapse, while its benefits vary across organs.

As described in Section 3.4, the wavelet filters are learned in a separate constrained filter-learning stage and remain fixed during architecture search and segmentation-network retraining. The architecture search optimizes the selection of their subband operations.

Table 7: Complete class-wise Dice scores (%) on the ten MSD tasks under aligned evaluation protocols. For multi-class tasks, Avg. denotes the arithmetic mean across the corresponding target classes. Best and second-best values are **bolded** and underlined, respectively.

Method	Brain Tumour				Heart	Liver		
	WT	TC	ET	Avg.	Atrium	Liver	Cancer	Avg.
MedNeXt	82.91	89.41	62.78	78.37	91.85	84.54	40.86	62.70
SwinUNETR	82.86	89.59	62.21	78.22	90.72	76.77	32.92	54.84
SegMamba	82.32	89.09	62.34	77.92	92.47	77.80	37.43	57.62
U-Mamba	81.14	89.26	61.20	77.20	83.22	81.33	35.53	58.43
SegMamba-V2	82.18	88.90	62.47	77.85	90.22	82.26	41.98	62.12
DiNTS	79.11	87.55	56.41	74.36	92.46	78.56	48.58	63.57
mCNA + Orth. LDW	**84.00**	**90.48**	**63.28**	**79.25**	**92.73**	**85.22**	**52.14**	**68.68**

Method	Hippocampus			Prostate			Lung
	Ant.	Post.	Avg.	Periph.	Transit.	Avg.	Cancer
MedNeXt	88.39	86.11	87.25	41.19	64.26	52.73	32.21
SwinUNETR	88.54	85.60	87.07	42.13	64.29	53.21	16.69
SegMamba	**88.94**	86.07	87.51	40.12	73.89	57.00	34.17
U-Mamba	87.63	86.49	87.06	46.18	69.49	57.83	37.60
SegMamba-V2	88.63	85.89	87.26	45.30	75.74	60.52	44.51
DiNTS	88.93	86.46	87.69	**57.95**	**85.65**	**71.80**	47.97
mCNA + Orth. LDW	**88.94**	**86.59**	**87.77**	45.35	82.51	63.93	**57.53**

Method	Pancreas			Hepatic Vessel			Spleen	Colon
	Pancreas	Cancer	Avg.	Vessel	Tumour	Avg.	Spleen	Cancer
MedNeXt	**77.33**	45.24	61.29	57.12	61.81	59.46	85.76	22.07
SwinUNETR	74.83	41.31	58.07	58.29	53.64	55.96	79.51	31.24
SegMamba	75.33	41.48	58.41	**59.65**	64.35	62.00	77.14	20.82
U-Mamba	74.91	43.43	59.17	58.11	66.68	62.39	86.26	22.62
SegMamba-V2	75.64	45.05	60.34	58.77	63.92	61.35	91.46	24.42
DiNTS	73.21	45.97	59.59	59.08	64.13	61.61	93.98	30.88
mCNA + Orth. LDW	76.54	**47.16**	**61.85**	59.03	**66.69**	**62.86**	**95.53**	**39.07**

E.2 Complete Class-Wise Results on MSD

The main paper reports the mean Dice score for each MSD task to provide a compact comparison across all ten tasks. Here, we provide the complete class-wise Dice scores for all multi-class tasks. For single-class tasks, the task-wise score is directly reported. All methods are retrained and evaluated using the aligned task-specific preprocessing and validation protocols described in Appendix D.5.

As shown in Table 7, mCNA + Orth. LDW achieves the highest task-level average Dice on nine of the ten MSD tasks. The exception is Task05 Prostate, where DiNTS achieves an average Dice of 71.80, compared with 63.93 for mCNA + Orth. LDW. At the class level, MedNeXt achieves the highest Dice on the pancreas class of Task07, while DiNTS achieves the highest Dice on both prostate subregions in Task05. For Task08 Hepatic Vessel, SegMamba achieves the highest vessel Dice, whereas mCNA + Orth. LDW achieves the highest tumour Dice and task-level average Dice. These class-wise results further show that the strongest method can vary across individual targets, while mCNA + Orth. LDW maintains strong task-level performance across the diverse MSD segmentation tasks.

Table 8: Extended component ablation on Synapse with a fixed clone number $N = 3$. Results are reported from a single training run.

NAS	CU	PAM	mDice (↑)	HD95 (↓)	Params (M)
✗	✗	✗	81.89	17.23	27.71
✗	✓	✗	82.46	11.87	43.64
✗	✓	✓	83.11	11.81	44.03
✓	✗	✗	82.58	15.54	26.54
✓	✓	✗	83.25	15.39	42.47
✓	✓	✓	**85.36**	**7.54**	42.86

Table 9: Effect of clone number N on Synapse. All configurations retain CU-based bottleneck processing; for $N = 1$, the incoming message is initialized as $m_0 = 0$ and no inter-clone message transfer occurs. Inference time and FLOPs are reported per input ROI. Results are reported from a single training run.

N	Dice (↑)	HD95 (↓)	Infer. (ms)	Params (M)	FLOPs (G)
1	82.15	17.71	6.37	14.29	72.08
2	84.39	12.52	12.46	28.57	143.80
3	**85.36**	**7.54**	22.89	42.86	216.76
4	84.03	10.98	30.93	57.14	287.61

E.3 Extended Component Ablation of NAS, CU, and PAM

We conduct a extended component ablation on Synapse to quantify the individual and combined effects of the searched cells (NAS), CU, and PAM. For configurations without NAS, we use a fixed hand-crafted cell architecture. Specifically, the DS-Cell adopts mean pooling for downsampling and standard $3 \times 3 \times 3$ convolutions for the remaining operations, whereas the US-Cell adopts transposed convolution for upsampling and standard convolutions for the remaining operations.

As shown in Table 8, the fixed hand-crafted baseline without CU or PAM achieves 81.89 mDice and 17.23 HD95. Adding CU improves these values to 82.46 and 11.87, respectively, and further adding PAM yields 83.11 mDice and 11.81 HD95.

Replacing the fixed cells with searched DS/US cells, without CU or PAM, yields 82.58 mDice and 15.54 HD95. Under the searched-cell configuration, adding CU increases mDice to 83.25 and slightly reduces HD95 to 15.39. Further adding PAM yields 85.36 mDice and 7.54 HD95, the best values of both metrics among the evaluated configurations.

These results show complementary performance gains from the searched cells, inter-clone feature interaction, and multi-scale skip weighting. The full configuration contains 42.86M parameters. The effect of different communication strategies and their associated model complexities is analyzed separately in Appendix E.7.

E.4 Effect of Clone Number

We evaluate clone numbers $N \in \{1, 2, 3, 4\}$ on Synapse to study the trade-off between cooperative refinement and computational complexity. All configurations use the same searched cell architecture, PAM, and CU-based bottleneck processing. For $N = 1$, the incoming communication state is initialized as $m_0 = 0$, such that no inter-clone message transfer occurs while retaining the same CU-based processing. As shown in Table 9, increasing the clone number from $N = 1$ to $N = 3$ improves Dice from 82.15 to 85.36 and reduces HD95 from 17.71 to 7.54. Further increasing the clone number to $N = 4$ decreases Dice to 84.03 and increases HD95 to 10.98, while inference time rises from 22.89 to 30.93 ms, parameter count from 42.86M to 57.14M, and FLOPs from 216.76G to 287.61G. Thus, the fourth clone incurs additional computational cost without improving either segmentation metric in this setting. We therefore adopt $N = 3$ as the default configuration, as it achieves the highest Dice and lowest HD95 among the evaluated clone numbers.

Table 10: Effect of U-Net depth on Synapse. Depth 4 provides a better Dice–parameter trade-off, while Depth 5 improves HD95 at substantially higher model complexity. Results are reported from a single training run.

Depth	Dice (↑)	HD95 (↓)	Params (M)
4	**85.36**	7.54	42.86
5	84.21	**5.43**	202.88

Table 11: Search-stage comparison and meta-node ablation under the search-once-transfer protocol. The reported cost corresponds to differentiable cell-level supernet optimization. Search FLOPs are reported in G and parameter counts in M. Architectures are searched on MSD Task07 Pancreas and retrained on Synapse.

Method	Nodes	Search Cost	Search FLOPs	Dice	HD95	Search Params	Retrain Params
DiNTS	–	0.56 d	1617.09	81.76	23.76	602.18	125.43
Ours	3	0.50 d	300.23	**85.36**	**7.54**	75.21	42.86
Ours	4	1.50 d	499.93	84.85	10.63	113.34	45.89

E.5 Effect of U-Net Depth

We evaluate two macro-depth configurations of the searched multi-clone communication U-Net backbone: Depth 4 and Depth 5. A deeper architecture enlarges the receptive field and improves multi-scale feature representation, but increases memory consumption and may make optimization more challenging in 3D settings.

Table 10 compares two macro-depth configurations of the multi-clone communication U-Net. Increasing the depth from 4 to 5 does not improve the overall overlap accuracy. The Dice score decreases from 85.36 to 84.21, while the parameter count increases substantially from 42.86M to 202.88M. This suggests that simply deepening the multi-clone communication U-Net significantly increases model complexity without improving region-level segmentation performance.

On the other hand, the HD95 score improves from 7.54 to 5.43, suggesting that the deeper model may reduce substantial boundary deviations, although this improvement comes with substantially higher complexity and lower Dice. Overall, these results reveal a trade-off between overlap accuracy, boundary quality, and model complexity. Although Depth 5 achieves better HD95, its substantially higher computational cost and lower Dice score make it less favorable as the default configuration. Therefore, we adopt Depth 4 in the main paper, as it provides a better balance between segmentation accuracy and parameter efficiency.

E.6 Search-stage Comparison and Meta-node Ablation

Following the search-once-transfer protocol used in DiNTS (He et al., 2021), we use MSD Task07 Pancreas as the proxy search task and transfer the searched architectures to Synapse by retraining from scratch. This protocol avoids dataset-specific search on Synapse and evaluates whether the searched architecture can generalize across abdominal segmentation datasets. For a controlled comparison, we perform architecture search for and retraining DiNTS under our aligned protocol rather than copying its official multi-GPU results. All methods use the same 80%/20% split for weight and architecture-parameter optimization, a search patch size of 96^3, batch size 1, 150 search epochs, and 200 iterations per epoch.

As shown in Table 11, our 3-meta-node search supernet reduces search-stage parameters from 602.18M to 75.21M and search-supernet FLOPs from 1617.09G to 300.23G compared with DiNTS. After transfer and retraining on Synapse, it improves Dice from 81.76 to 85.36 and reduces HD95 from 23.76 to 7.54. These single-run results are distinct from the three-run statistics reported in Table 1.

Increasing the number of meta-nodes from 3 to 4 raises search FLOPs from 300.23G to 499.93G and search-stage parameters from 75.21M to 113.34M. However, Dice decreases from 85.36 to 84.85,

Table 12: Comparison of different inter-clone communication strategies on Synapse.

Strategy	Dice (↑)	HD95 (↓)	Params (M)	FLOPs (G)
No Comm.	82.56	17.99	26.92	208.60
Add	83.11	13.69	26.92	208.61
Concat+Fuse	83.36	17.51	37.54	214.05
CU	**85.36**	**7.54**	42.86	216.76

while HD95 increases from 7.54 to 10.63. Search cost also rises from 0.50 to 1.50 days. Therefore, we adopt 3 meta-nodes as the default setting, as this configuration achieves better Dice and HD95 at lower search cost in the evaluated comparison.

E.7 Ablation of Inter-Clone Communication

We compare four inter-clone communication strategies to evaluate the effectiveness of the proposed Communication Unit (CU). The first setting removes inter-clone information propagation, such that each clone performs segmentation independently. The second setting directly adds the features transmitted from the preceding clone. The third setting concatenates the transmitted and current features, followed by a fusion operation. The final setting uses the proposed CU to adaptively integrate inter-clone information.

As shown in Table 12, the No Comm. setting achieves 82.56 Dice and 17.99 HD95. Direct addition improves Dice to 83.11 and reduces HD95 to 13.69 with essentially unchanged parameter count and FLOPs, indicating that inter-clone information propagation can improve segmentation without relying on increased model capacity.

Concat+Fuse increases the parameter count from 26.92M to 37.54M compared with No Comm., but improves Dice by only 0.80 percentage points and reduces HD95 by 0.48. In contrast, the proposed CU achieves the highest Dice of 85.36 and the lowest HD95 of 7.54, with 42.86M parameters and 216.76G FLOPs. Compared with Concat+Fuse, CU further improves Dice by 2.00 percentage points and reduces HD95 by 9.97, while increasing parameters by 5.32M and FLOPs by 2.71G.

These results show that CU outperforms the evaluated communication alternatives under the current configurations. Because the variants are not strictly capacity-matched, the observed improvement may reflect both gated message passing and differences in model capacity.

F Additional 2D Segmentation Experiments

To further demonstrate the feasibility and generality of the proposed multi-clone communication design, we additionally conduct experiments on 2D medical image segmentation benchmarks. These experiments are complementary to the main 3D volumetric segmentation experiments and are used to examine whether the proposed design remains effective under 2D segmentation settings. Specifically, we evaluate the 2D variant of mCNA on CHAOS CT/MR, BAGLS, and KVASIR, covering liver, multi-organ abdominal, laryngeal, and polyp segmentation tasks. We compare our method with representative manually designed networks, NAS-based segmentation methods, and task-specific state-of-the-art methods using region-level Dice-based metrics, including mDice for the binary datasets and organ-wise Dice for CHAOS MR-only, together with boundary-aware metrics, including Boundary F1 Score (BF1) and Hausdorff Distance (HD).

For the CHAOS dataset, the CT-only category contains CT images of 40 different patients with 512×512 resolution. In total, 1,367 slices are provided for training and 1,408 slices are used for testing. For the MRI-only category, 120 DICOM image sets are selected from T1-DUAL in-phase, T1-DUAL out-phase, and T2-SPIR sequences. The image resolution is 256×256, and the number of slices ranges from 26 to 50. In total, 1,594 slices are used for training and 1,537 slices are used for testing. For datasets that require only binary outputs, we search the mCNA architecture on the CHAOS CT dataset and directly transfer, retrain, and fine-tune it on the other binary segmentation datasets. For the CHAOS MR-only dataset, which requires four abdominal organ outputs, a separate architecture search is performed for mCNA to accommodate the four-label segmentation setting.

F.1 Region-level Comparison on 2D Benchmarks

Table 13 presents the mDice comparisons on CHAOS CT, BAGLS, and KVASIR.

On CHAOS CT, mCNA achieves an mDice of 96.8%, outperforming all listed non-NAS baselines and remaining comparable to NAS-Unet and MS-NAS, both of which achieve 97.4%. mCNA + Orth. LDW further improves the mDice to 97.6%, achieving the best result among all compared methods.

On BAGLS, mCNA substantially outperforms U-Net and all compared NAS-based methods, while mCNA + Orth. LDW further improves the mDice from 92.1% to 92.4%.

On KVASIR, mCNA also outperforms the listed task-specific polyp segmentation methods, including PVT, XBound-F, PraNet, TransFuse, BAT, ACSNet, and FTMF-Net. mCNA + Orth. LDW further improves the mDice from 86.9% to 87.5%.

Table 14 presents the organ-wise Dice scores and parameter counts on the CHAOS MR-only dataset. Even without LDW pooling, mCNA outperforms all compared NAS-based methods across the four abdominal organs. Adding LDW pooling further improves the Dice scores to 96.8%, 93.2%, 95.3%, and 92.5% for the liver, left kidney, right kidney, and spleen, respectively.

Compared with U-Net, the largest improvement is observed for the left kidney, where Dice increases from 73.8% to 93.2%, corresponding to an improvement of 19.4 percentage points. Overall, mCNA + LDW achieves the highest Dice score on all four organs among the compared methods.

F.2 Boundary-level Comparison on 2D Benchmarks

Table 15 presents a comparative analysis of our method against other state-of-the-art approaches across the CHAOS CT, BAGLS, and KVASIR datasets based on BF1 and HD. Notably, the images in the KVASIR dataset have significantly larger dimensions compared with the others, resulting in proportionally higher HD values in the KVASIR category. Among the NAS-based baselines, MS-NAS achieves the highest BF1 on CHAOS CT and BAGLS, while AutoDeepLab achieves the highest BF1 on KVASIR. For HD, AutoDeepLab performs best on CHAOS CT, whereas MS-NAS performs best on BAGLS and KVASIR. In contrast, mCNA + Orth. LDW achieves the best BF1 and HD results across all three datasets, demonstrating improved boundary accuracy and robustness.

Table 16 further presents the comparison on the CHAOS MR-only dataset using boundary-aware metrics. Even under these boundary-focused evaluations, mCNA + LDW consistently outperforms all NAS-based baselines. Collectively, these results provide additional empirical evidence for the effectiveness and robustness of mCNA across the evaluated medical image segmentation benchmarks.

Overall, these 2D experiments provide additional evidence that the proposed multi-clone communication design is feasible beyond 3D volumetric segmentation. The results show consistent improvements across CT, MR, laryngeal, and polyp segmentation tasks. Together with the main 3D experiments, these comparisons further support the effectiveness of communication-enabled cooperative refinement, path-aware feature fusion, and LDW-based structure-preserving downsampling across different medical segmentation settings.

G Additional qualitative results

This section provides additional qualitative comparisons on Synapse and MSD. For Synapse, predictions are generated using sliding-window inference with a 128^3 window and 50% overlap, following the setting used in the main quantitative evaluation. The compared methods include 3D U-Net, SegResNet, 3D UX-Net, MedNeXt, UNETR, SwinUNETR, SegMamba, U-Mamba, SegMamba-V2, and DiNTS. For MSD, predictions are generated using sliding-window inference with a 96^3 window and 50% overlap. The compared methods include MedNeXt, SwinUNETR, SegMamba, U-Mamba, and SegMamba-V2. For all MSD qualitative comparisons, the proposed method refers to mCNA + Orth. LDW, which is the final configuration used in the quantitative evaluation. These visualizations complement the quantitative results by highlighting boundary quality, anatomical completeness, local structural consistency, and small-target localization across different segmentation tasks.

Table 13: Performance comparisons on CHAOS CT, BAGLS, and KVASIR. All results are reported in percentage.

(a) CHAOS CT

Method	mDice (%)
U-Net (Ronneberger et al., 2015)	95.0 ± 0.7
PSP-Net (Zhao et al., 2017)	93.8
MANet (Fan et al., 2020b)	94.1
DeepLabV3+ (Chen et al., 2018)	95.0
AttU-Net (Oktay et al., 2018)	96.7
DeTTA (Wen et al., 2024)	96.7
Converted3DNet (Hassanzadeh et al., 2020)	88.9
NAS-Unet (Weng et al., 2019)	97.4 ± 0.7
AutoDeepLab (Liu et al., 2019)	96.3 ± 0.7
MS-NAS (Yan et al., 2020)	97.4 ± 0.6
mCNA	96.8 ± 0.6
mCNA + Orth. LDW	$\mathbf{97.6 \pm 0.5}$

(b) BAGLS

Method	mDice (%)
U-Net (Ronneberger et al., 2015)	80.9 ± 0.6
NAS-Unet (Weng et al., 2019)	81.7 ± 0.7
AutoDeepLab (Liu et al., 2019)	87.8 ± 0.6
MS-NAS (Yan et al., 2020)	88.3 ± 0.5
mCNA	92.1 ± 0.5
mCNA + Orth. LDW	$\mathbf{92.4 \pm 0.4}$

(c) KVASIR

Method	mDice (%)
U-Net (Ronneberger et al., 2015)	63.6 ± 0.7
PVT (Dong et al.)	72.0
XBound-F (Wang et al., 2023)	78.5
PraNet (Fan et al., 2020a)	78.7
TransFuse (Zhang et al., 2021)	78.8
BAT (Wang et al., 2021b)	81.0
ACSNet (Zhang et al., 2020b)	83.5
FTMF-Net (Liu et al., 2023)	86.3
mCNA	86.9 ± 0.6
mCNA + Orth. LDW	$\mathbf{87.5 \pm 0.5}$

G.1 Qualitative results on Synapse

Figure 6 presents qualitative segmentation results on Synapse. Compared with multiple methods, including 3D U-Net, SegResNet, 3D UX-Net, MedNeXt, UNETR, SwinUNETR, U-Mamba, SegMamba, SegMamba-V2, and DiNTS, our method produces more spatially coherent organ predictions and better preserves local anatomical details. In the zoomed regions, competing methods often exhibit fragmented organ masks, incomplete small-structure segmentation, or less accurate boundary delineation. In contrast, the proposed method yields smoother and more complete segmentation results across multiple organs, which is consistent with the improved Dice and HD95 scores reported in the main paper.

Table 14: Dice score and parameter comparisons with state-of-the-art methods on the CHAOS MR-only dataset. Dice scores are reported in percentage.

Method	Liver Dice (%)	Left Kidney Dice (%)	Right Kidney Dice (%)
U-Net (Ronneberger et al., 2015)	86.8 ± 0.5	73.8 ± 0.6	76.2 ± 0.5
mCNA	95.9 ± 0.4	92.0 ± 0.5	95.0 ± 0.4
mCNA + LDW	$\mathbf{96.8 \pm 0.4}$	$\mathbf{93.2 \pm 0.5}$	$\mathbf{95.3 \pm 0.5}$
NAS-Unet (Weng et al., 2019)	87.2 ± 0.4	75.0 ± 0.6	77.7 ± 0.5
AutoDeepLab (Liu et al., 2019)	88.1 ± 0.6	75.0 ± 0.7	75.7 ± 0.7
MS-NAS (Yan et al., 2020)	82.6 ± 0.4	81.9 ± 0.6	81.1 ± 0.6
BiX-Net (Wang et al., 2021c)	89.8	82.1	82.7
ECT-NAS (Xu & Quan, 2021)	94.1	88.7	87.4
EU-Net (Yu et al., 2023)	90.5	90.0	94.7

Method	Spleen Dice (%)	Params (M)
U-Net (Ronneberger et al., 2015)	74.5 ± 0.6	7.14
mCNA	92.1 ± 0.4	46.17
mCNA + LDW	$\mathbf{92.5 \pm 0.5}$	52.29
NAS-Unet (Weng et al., 2019)	75.8 ± 0.6	10.26
AutoDeepLab (Liu et al., 2019)	75.5 ± 0.6	36.84
MS-NAS (Yan et al., 2020)	74.0 ± 0.5	21.17
BiX-Net (Wang et al., 2021c)	76.5	8.56
ECT-NAS (Xu & Quan, 2021)	90.1	–
EU-Net (Yu et al., 2023)	91.7	–

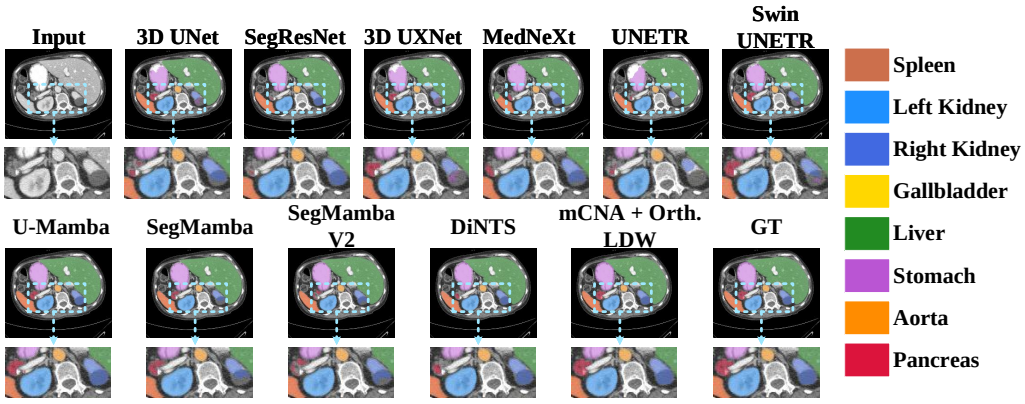

Figure 6: Qualitative segmentation results on Synapse. The zoomed regions highlight local anatomical details and boundary consistency across multiple organs.

G.2 Qualitative results on MSD Task01 BrainTumour

Figure 7 presents qualitative segmentation results on MSD Task01 BrainTumour. In the visualization, we show the original annotation categories, including edema (E), non-enhancing tumour (N), and enhancing tumour (H). For the quantitative evaluation reported in the main paper, these original labels are converted into the BraTS-style composite regions:

$$\mathrm{WT} = E \cup N \cup H, \quad \mathrm{TC} = N \cup H, \quad \mathrm{ET} = H,$$

Table 15: Performance comparisons against state-of-the-art methods on CHAOS CT, BAGLS, and KVASIR using Boundary F1 Score (BF1) and Hausdorff Distance (HD).

Methods	CHAOS CT	
	BF1 (↑)	HD (↓)
U-Net (Ronneberger et al., 2015)	0.488 ± 0.019	19.46 ± 1.67
NAS-Unet (Weng et al., 2019)	0.498 ± 0.018	17.08 ± 1.43
AutoDeepLab (Liu et al., 2019)	0.570 ± 0.017	10.62 ± 0.94
MS-NAS (Yan et al., 2020)	0.619 ± 0.017	15.25 ± 1.18
mCNA + LDW	0.638 ± 0.017	9.75 ± 1.05
mCNA + Orth. LDW	**0.654** ± 0.016	**9.56** ± 0.96

Methods	BAGLS	
	BF1 (↑)	HD (↓)
U-Net (Ronneberger et al., 2015)	0.785 ± 0.021	25.24 ± 0.98
NAS-Unet (Weng et al., 2019)	0.889 ± 0.022	20.83 ± 0.95
AutoDeepLab (Liu et al., 2019)	0.874 ± 0.017	18.08 ± 0.73
MS-NAS (Yan et al., 2020)	0.893 ± 0.019	9.66 ± 0.71
mCNA + LDW	0.916 ± 0.017	9.42 ± 0.72
mCNA + Orth. LDW	**0.938** ± 0.018	**9.15** ± 0.67

Methods	KVASIR	
	BF1 (↑)	HD (↓)
U-Net (Ronneberger et al., 2015)	0.493 ± 0.018	43.50 ± 3.03
NAS-Unet (Weng et al., 2019)	0.502 ± 0.018	39.41 ± 2.86
AutoDeepLab (Liu et al., 2019)	0.527 ± 0.016	35.94 ± 2.67
MS-NAS (Yan et al., 2020)	0.518 ± 0.017	33.24 ± 2.83
mCNA + LDW	0.557 ± 0.018	32.06 ± 2.78
mCNA + Orth. LDW	**0.587** ± 0.016	**30.85** ± 2.53

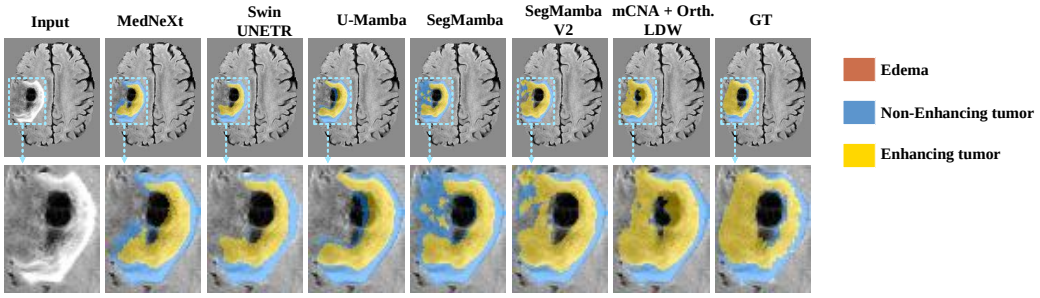

Figure 7: Qualitative comparison on MSD Task01 BrainTumour. The proposed method produces a more complete whole-tumour prediction, but also shows a small false-positive enhancing-tumour response in the central dark region.

where WT, TC, and ET denote whole tumour, tumour core, and enhancing tumour, respectively.

As shown in the zoomed region, the proposed mCNA + Orth. LDW produces a more complete tumour prediction than several competing methods, especially for the overall tumour extent. However, it also introduces a small false-positive enhancing-tumour response in the central dark region. This indicates that the proposed method better preserves tumour completeness, while fine-grained tumour subregion classification remains challenging in ambiguous internal regions.

Table 16: Performance comparisons against different state-of-the-art methods on the CHAOS MR-only dataset using Boundary F1 Score (BF1) and Hausdorff Distance (HD).

Methods	Liver		Left Kidney	
	BF1 (↑)	HD (↓)	BF1 (↑)	HD (↓)
U-Net (Ronneberger et al., 2015)	0.502 ± 0.019	19.82 ± 1.35	0.463 ± 0.026	26.82 ± 1.98
NAS-Unet (Weng et al., 2019)	0.521 ± 0.015	17.91 ± 1.22	0.483 ± 0.025	23.29 ± 1.94
AutoDeepLab (Liu et al., 2019)	0.564 ± 0.019	12.93 ± 1.36	0.654 ± 0.019	7.59 ± 0.56
MS-NAS (Yan et al., 2020)	0.724 ± 0.016	18.37 ± 1.32	0.581 ± 0.016	14.60 ± 0.82
mCNA + LDW	**0.891** ± 0.014	**11.78** ± 1.23	**0.705** ± 0.017	**6.68** ± 0.52

Methods	Right Kidney		Spleen	
	BF1 (↑)	HD (↓)	BF1 (↑)	HD (↓)
U-Net (Ronneberger et al., 2015)	0.565 ± 0.021	13.64 ± 1.54	0.382 ± 0.022	24.35 ± 1.54
NAS-Unet (Weng et al., 2019)	0.588 ± 0.021	12.28 ± 1.42	0.391 ± 0.018	22.17 ± 1.51
AutoDeepLab (Liu et al., 2019)	0.630 ± 0.021	13.66 ± 0.78	0.502 ± 0.012	8.96 ± 0.59
MS-NAS (Yan et al., 2020)	0.626 ± 0.018	11.28 ± 0.84	0.568 ± 0.022	18.74 ± 0.95
mCNA + LDW	**0.725** ± 0.019	**10.85** ± 0.84	**0.627** ± 0.014	**8.19** ± 0.58

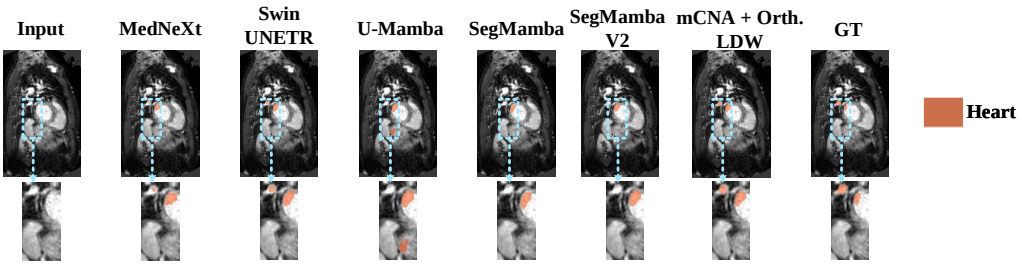

Figure 8: Qualitative comparison on MSD Task02 Heart. The zoomed region highlights a subtle local structure where the proposed mCNA + Orth. LDW produces a more accurate prediction, while several competing methods show missing segmentation or false-positive responses.

G.3 Qualitative results on MSD Task02 Heart

Figure 8 presents qualitative segmentation results on MSD Task02 Heart. In the highlighted region, the proposed mCNA + Orth. LDW produces a more accurate prediction for the small target area. In contrast, several competing methods exhibit missing segmentation or false-positive responses around this subtle local structure. This example shows the advantage of the proposed framework in preserving fine anatomical details during volumetric segmentation.

G.4 Qualitative results on MSD Task03 Liver

Figure 9 presents qualitative segmentation results on MSD Task03 Liver. The zoomed region shows that the proposed mCNA + Orth. LDW more accurately localizes the small cancer region while preserving a more complete liver structure. In contrast, several competing methods produce less precise cancer predictions or show incomplete liver boundaries around the highlighted area.

G.5 Qualitative results on MSD Task04 Hippocampus

Figure 10 presents qualitative segmentation results on MSD Task04 Hippocampus. Unlike lesion segmentation tasks, this task focuses on separating the anterior and posterior parts of the same

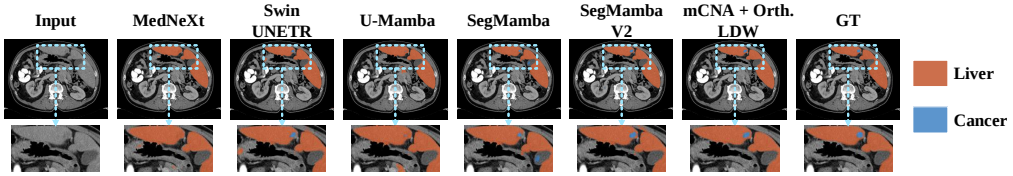

Figure 9: Qualitative comparison on MSD Task03 Liver. The proposed method improves small cancer localization and preserves a more complete liver structure.

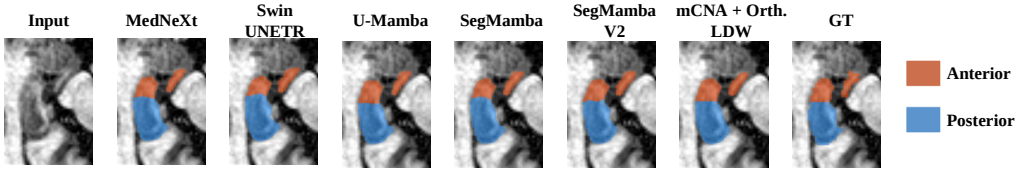

Figure 10: Qualitative comparison on MSD Task04 Hippocampus. The proposed method provides more accurate boundary delineation between the anterior and posterior hippocampus regions.

anatomical structure. Although most methods produce reasonable overall hippocampus predictions, the proposed mCNA + Orth. LDW provides a more accurate delineation around the boundary between the anterior and posterior regions. This suggests that the proposed framework can better preserve local structural details even when the target classes belong to adjacent subregions of the same organ.

G.6 Qualitative results on MSD Task05 Prostate

Figure 11 presents qualitative segmentation results on MSD Task05 Prostate. This task focuses on segmenting two internal prostate regions, namely the peripheral zone and the transitional zone. Compared with the competing methods, mCNA + Orth. LDW produces smoother boundaries and a more coherent prostate-zone structure that is better aligned with the ground truth.

G.7 Qualitative results on MSD Task06 Lung

Figure 12 presents qualitative segmentation results on MSD Task06 Lung. This task focuses on segmenting lung cancer regions, where surrounding low-contrast tissues and noisy responses can easily lead to false-positive predictions. As shown in the zoomed region, several competing methods are affected by nearby noisy structures and produce extra segmented regions. In contrast, the proposed mCNA + Orth. LDW better suppresses these false positives and identifies the true target region more accurately.

G.8 Qualitative results on MSD Task07 Pancreas

Figure 13 presents qualitative segmentation results on MSD Task07 Pancreas. This task requires segmenting both the pancreas and pancreatic cancer regions, where the tumour area is often small and difficult to localize. The proposed method further preserves the elongated pancreas structure and provides more consistent local segmentation around the highlighted region. These visual observations complement the higher task-level mean Dice and cancer Dice reported in the main paper, while MedNeXt retains the highest pancreas-class Dice.

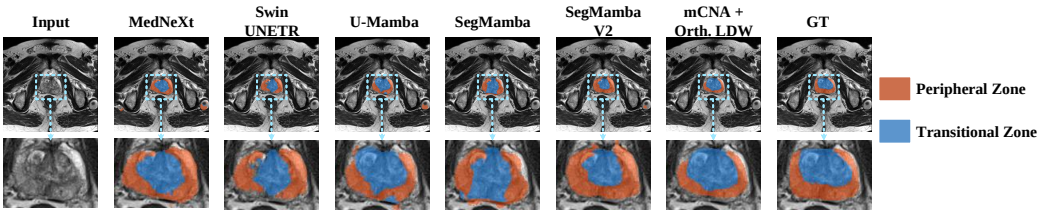

Figure 11: Qualitative comparison on MSD Task05 Prostate. In the illustrated case, mCNA + Orth. LDW provides smoother boundaries and a more coherent structure for peripheral-zone and transitional-zone segmentation.

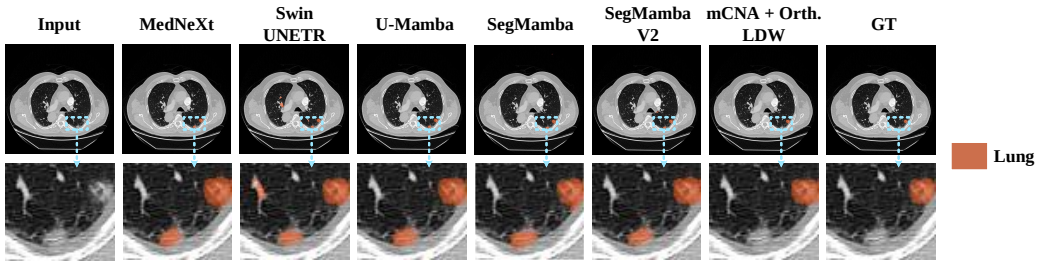

Figure 12: Qualitative comparison on MSD Task06 Lung. The proposed method better suppresses false-positive responses from nearby noisy structures and more accurately identifies the target lung cancer region.

G.9 Qualitative results on MSD Task08 Hepatic Vessel

Figure 14 presents qualitative segmentation results on MSD Task08 Hepatic Vessel. This task is challenging because hepatic vessels are thin, fragmented, and locally similar to surrounding background structures. As shown in the zoomed region, several competing methods produce false-positive vessel predictions in background areas or generate less coherent vessel structures. In contrast, the proposed mCNA + Orth. LDW better suppresses background responses while preserving the true vessel and tumour regions.

G.10 Qualitative results on MSD Task09 Spleen

Figure 15 presents qualitative segmentation results on MSD Task09 Spleen. Although the highlighted spleen region is relatively small in this example, the proposed mCNA + Orth. LDW is able to more completely identify the target region and produce a prediction that is better aligned with the ground truth. In contrast, several competing methods yield less complete segmentation in the zoomed area. This example further shows the advantage of the proposed framework in preserving fine structural information for small target regions.

G.11 Qualitative results on MSD Task10 Colon

Figure 16 presents qualitative segmentation results on MSD Task10 Colon. As shown in the zoomed region, several competing methods are affected by nearby noisy structures and produce false-positive responses in the upper-right area. In contrast, the proposed mCNA + Orth. LDW more accurately distinguishes foreground from background and avoids these incorrect predictions. This example further demonstrates the robustness of the proposed framework in suppressing noise interference and localizing the true target region.

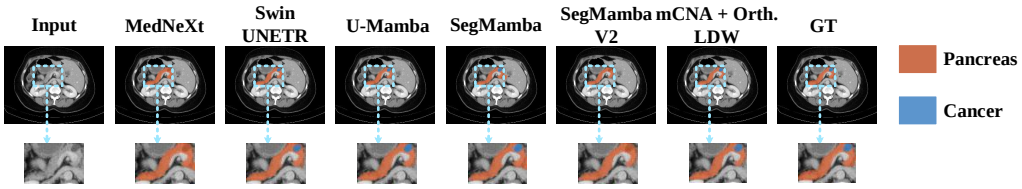

Figure 13: Qualitative comparison on MSD Task07 Pancreas. The proposed method better preserves the elongated pancreas structure and improves local segmentation consistency.

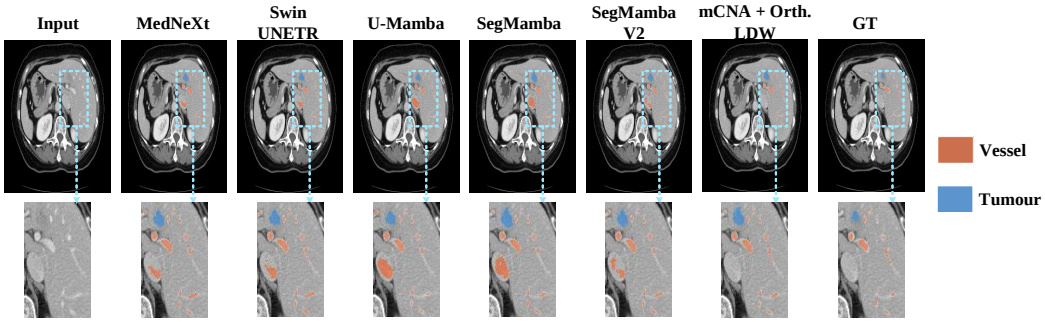

Figure 14: Qualitative comparison on MSD Task08 Hepatic Vessel. The proposed method better suppresses false-positive vessel responses in background regions while preserving fine vessel and tumour structures.

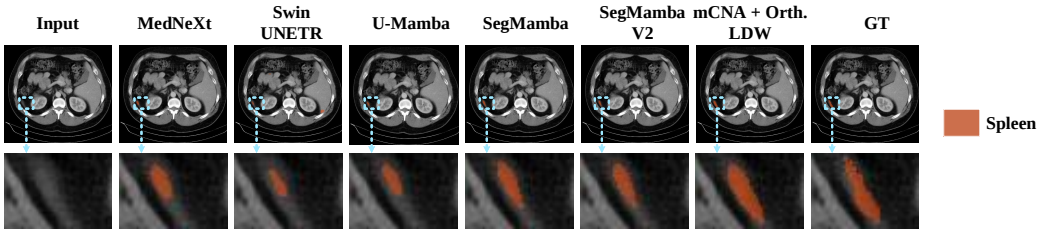

Figure 15: Qualitative comparison on MSD Task09 Spleen. The proposed method more completely identifies the spleen region in the highlighted area and produces a prediction closer to the ground truth.

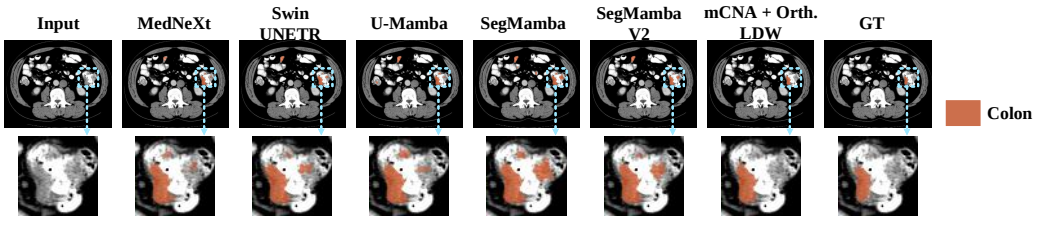

Figure 16: Qualitative comparison on MSD Task10 Colon. The proposed method better suppresses false-positive responses in the noisy upper-right region and more accurately distinguishes foreground from background.