NEURAL ARCHITECTURE DISTRIBUTIONS: A NEW PARADIGM FOR STOCHASTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic segmentation seeks to represent multiple plausible masks for a single image, a requirement in safety- and quality-critical applications such as medical imaging and building defect inspection. Most existing methods introduce stochasticity by injecting continuous latent variables or by following iterative denoising trajectories, making their stochastic sources difficult to search or audit. We study architecture distributions as a stochastic source for segmentation: instead of sampling a latent variable or noise, we sample a discrete architecture from a learned distribution over operator choices at multiple searchable positions in a segmentation backbone. Each sampled architecture yields one mask through the selected active path, so inference depends on the executed subnet rather than the complete candidate bank. This approach supports architectural provenance, since each output corresponds to a specific architecture configuration. To reduce collapse toward averaged masks, we train with set-level supervision by matching architecture-sampled predictions to the annotation set using an IoU-based energy-distance surrogate. We construct the candidate bank with evolutionary search, making the support of the stochastic source optimizable before distribution learning. NADS improves distribution matching and hypothesis coverage on LIDC-IDRI, and generalizes to additional stochastic benchmarks (Hippocampus, QUBIQ Prostate) as well as deterministic segmentation tasks (Cityscapes, Crack500). The resulting formulation learns an input-conditioned architecture distribution and produces multi-annotator output diversity through architecture sampling with set-level supervision.

1 INTRODUCTION

Semantic segmentation underlies many safety- and quality-critical vision systems, including medical image analysis, building and infrastructure inspection, and transportation-related perception (Kohl et al., 2018). In these applications the target mask is often ambiguous: multiple segmentations can be plausible for the same input, from different experts' annotations, due to limited image evidence, weak boundaries, or inherent subjectivity among annotators (Zbinden et al., 2023; Monteiro et al., 2020). Beyond producing a single point estimate, it is therefore desirable to model a conditional distribution over masks given an image, so that the model can generate multiple plausible hypotheses and support uncertainty-aware decision making.

Most segmentation pipelines are built upon deterministic convolutional backbones and output one mask per input. Architectures such as U-Net and its variants have become standard due to their empirical performance and inference efficiency (Ronneberger et al., 2015; Zhou et al., 2018; Cao et al., 2022). However, when ambiguity exists, pixel-wise supervision with a single annotation encourages the model to predict an "average" solution, which may not correspond to any valid annotation mode. This limitation motivates stochastic segmentation, where the goal is to approximate the distribution of plausible masks rather than a single output.

Our motivation is that many segmentation ambiguities are not arbitrary pixel-wise noise, but discrete structural alternatives: annotators may disagree on whether a weak boundary should be included, whether a small uncertain component connects to the main region, or how far an object extends into low-contrast areas. Such alternatives correspond to different structural priors over boundary sharpness, receptive field, local smoothness, and shape connectivity. Candidate

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.

operators in an architecture bank encode these biases through their kernel size, dilation, depth, residual/MBConv/ConvNeXt-style transformations, and identity paths. Sampling an architecture therefore samples an executable decision function with a particular structural bias, rather than only perturbing the feature representation of a fixed decoder. This motivates using an input-conditioned architecture distribution to represent annotator-dependent hypotheses.

Existing stochastic segmentation approaches introduce randomness through latent variables or iterative generative processes. A representative line of work couples a segmentation backbone with conditional latent-variable models, such as the Probabilistic U-Net (Kohl et al., 2018), to sample diverse outputs by drawing a latent code. Other approaches model a structured distribution in the logit space, such as Stochastic Segmentation Networks (SSN) (Monteiro et al., 2020). Diffusion and score-based generative models have also been explored for image generation and have inspired stochastic prediction mechanisms that rely on iterative denoising or numerical solvers (Rahman et al., 2023; Zbinden et al., 2023). To increase the expressiveness of latent distributions, several works augment variational posteriors using normalizing flows, which transform simple distributions through sequences of invertible mappings (Li et al., 2025; Papamakarios et al., 2021; Valiuddin et al., 2021).

Despite their progress, these latent-variable and diffusion-style methods have two practical drawbacks in safety-critical settings. First, improving the expressiveness of the stochastic source often increases inference cost: diffusion models typically require multiple sampling steps, and flow-augmented variational models add extra transformations to obtain a more flexible posterior (Ho et al., 2020; Papamakarios et al., 2021). Second, sample traceability is limited: because the randomness is a continuous noise/code or denoising trajectory, individual masks do not correspond to a discrete, human-interpretable generation pathway. Yet recording how a specific hypothesis is produced can support auditing and diagnosis when predictions are used for decision support.

In this work, we model stochastic segmentation as a distribution over architectures rather than injecting stochasticity into features or latent variables. We insert a set of searchable positions into a backbone and define a categorical architecture distribution over candidate operators at each position. This turns the stochastic source into a discrete, searchable support: the candidate bank can be shaped by evolutionary search before the input-conditioned distribution is learned with set-level supervision. At inference time, one segmentation sample is obtained by sampling an architecture and executing its active path. This design has two advantages. First, inference is determined by the selected active path rather than by the complete stored candidate bank, because only one operator per searchable position is executed for each sample. Second, every predicted mask is associated with a discrete architecture code, providing an auditable generation trace.

To learn a distribution of plausible masks, we adopt set-level supervision that matches a set of predicted samples to the set of annotations, and we optimize a distributional objective based on the generalized energy distance (GED) as popularized for stochastic segmentation evaluation (Kohl et al., 2018). We further construct a performance-oriented candidate operator bank using an aging-evolution style search procedure (Real et al., 2019), and train the final architecture distribution model with weight inheritance and stochastic architecture sampling. In summary, our contributions are:

- We introduce architecture distributions as a stochastic source for segmentation, where randomness is realized by sampling discrete architectures rather than injecting latent noise.
- We show that the support of this stochastic source can be optimized by evolutionary construction of the candidate bank, providing a searchable alternative to fixed latent/noise sources.
- We develop a set-level objective based on an energy-distance surrogate to prevent distribution degradation (collapse or averaging) and to better match multi-annotator labels.
- We demonstrate that architecture sampling improves distribution matching while maintaining inference time comparable to representative stochastic segmentation baselines and providing architectural provenance.

2 Related Work

2.1 Stochastic Segmentation

Stochastic segmentation aims to model multiple plausible masks for the same input rather than a single deterministic output. This setting is common when boundaries are visually ambiguous, when annotators disagree, or when the target concept is subjective. Early uncertainty baselines rely on approximate Bayesian inference such as Monte-Carlo dropout or ensembles (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017), but are not designed to match the empirical distribution induced by multi-annotator labels and may still favor an "average" solution.

A major line formulates the task as conditional generative modeling with latent variables. Probabilistic U-Net couples a U-Net backbone with a conditional variational autoencoder (CVAE) to sample image-conditioned segmentations and evaluates distribution matching with set-level metrics such as GED (Kohl et al., 2018). PHiSeg adds hierarchical latent variables to capture ambiguity across spatial scales, addressing the limitation that a single global latent may be ignored or fail to express localized variations (Baumgartner et al., 2019; Kohl et al., 2019). Follow-up works use flow-based methods to better align the prior and posterior latent distributions (Selvan et al., 2020; Valiuddin et al., 2021; Bhat et al., 2022; De Sousa Ribeiro et al., 2025). A second direction represents stochasticity in the logit space. SSN models spatially-correlated segmentation logits and draws coherent masks without an explicit latent-image decoder (Monteiro et al., 2020), while diffusion models generate masks via iterative denoising (Rahman et al., 2023; Zbinden et al., 2023; Baru et al., 2026), which typically requires many steps and is costly at inference.

2.2 Architecture Distributions

Learning distributions over discrete architectural choices has been studied in neural architecture search (NAS). DARTS (Liu et al., 2019) relaxes discrete operator selection into continuous mixtures for gradient-based optimization; SNAS (Xie et al., 2019) models operator choices as categorical variables with reparameterized sampling; ENAS (Pham et al., 2018) trains a weight-shared over-parameterized network whose sub-networks inherit shared weights; and Once-for-All (Cai et al., 2020) trains a single network supporting many specializable subnetworks. Closer to our motivation, a separate line exploits architecture *multiplicity* rather than a single searched model. Neural Ensemble Search (Zaidi et al., 2021) searches diverse architectures whose ensemble improves uncertainty and robustness to distribution shift, while BatchEnsemble (Wen et al., 2020) and MIMO (Havasi et al., 2021) produce several predictions from one weight-shared network. These establish architecture diversity as a source of predictive diversity, but target classification-style uncertainty with an unconditional ensemble and are not trained to match pixel-level annotation sets.

These NAS methods learn architecture distributions to search for one model under accuracy/efficiency constraints, and the ensemble methods form an unconditional set for uncertainty; in contrast, stochastic segmentation learns a conditional output distribution over multiple plausible masks. This motivates using architecture sampling not only as a search tool but as a mechanism for parameterizing and sampling the model's conditional prediction distribution. Unlike the ensemble-based methods, our architecture distribution is input-conditioned and trained with set-level (energy-distance) supervision to match the multi-annotator label set.

3 Methodology

3.1 Preliminaries

We study stochastic segmentation where multiple segmentation masks can be valid for the same input image. Let $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ denote an input image. For each image, we observe a *set* of M binary annotations $Y(x) = \{y^{(m)}\}_{m=1}^{M}$ with $y^{(m)} \in \{0,1\}^{H \times W}$, provided by different annotators or repeated annotations. This setting corresponds to an unknown conditional distribution $p(y \mid x)$ rather than a single deterministic target. Our goal is to learn a model that can produce multiple plausible segmentations for a given input. At inference time, the model should generate a set of K predictions ($K \geq M$), $\hat{\mathcal{Y}}(x) = \{\hat{y}^{(1)}, \ldots, \hat{y}^{(K)}\}$, such that the distribution of $\hat{\mathcal{Y}}(x)$ matches the annotation distribution represented by $\mathcal{Y}(x)$.

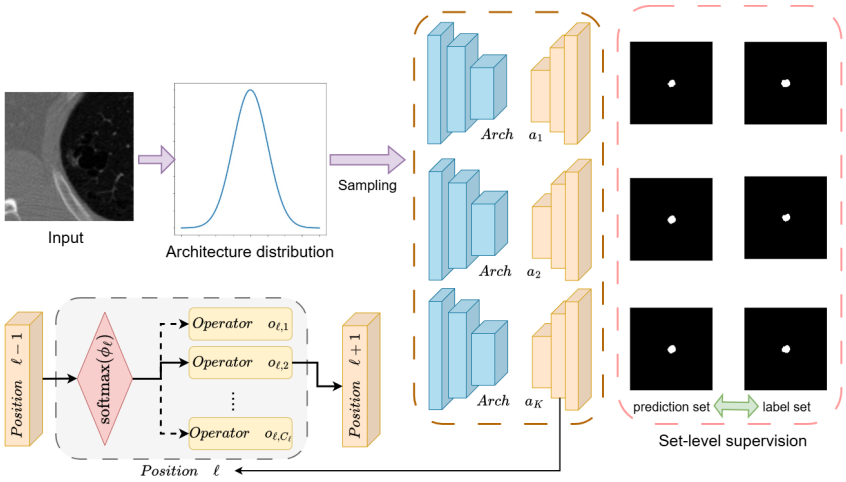


Figure 1: Overview of the NADS framework. Given an input, the model predicts an input-conditioned distribution over discrete architectures and samples K architectures from it. Each sampled architecture instantiates a single-path subnet to produce one segmentation hypothesis, and the prediction set is trained with set-level supervision against the multi-annotator label set.

Unlike latent-variable approaches (CVAE or diffusion) that inject random noise into inputs or hidden states, we represent stochasticity as a distribution over architectures. We define a discrete architecture variable a with a learnable, input-conditioned distribution $q_\psi(a \mid x)$. Each sample $a \sim q_\psi(a \mid x)$ instantiates a concrete architecture within a weight-inheritance network that predicts through

$$\hat{p}(x; a) = \sigma(f_\theta(x; a)), \qquad \hat{y}(x; a) = \mathbb{I}[\hat{p}(x; a) > 0.5], \tag{1}$$

where $f_\theta(\cdot; a)$ denotes the segmentation network instantiated by architecture a, θ denotes the shared weights, and $\sigma(\cdot)$ is the sigmoid function. For multi-class segmentation, we replace the sigmoid head with a softmax head. A set of predictions is obtained by sampling architectures multiple times ($a^{(k)} \sim q_\psi(a \mid x)$, $\hat{y}^{(k)} = \hat{y}(x; a^{(k)})$, $k{=}1, \ldots, K$), so the model outputs a segmentation distribution through architecture sampling while keeping the underlying weights inherited and the forward pass standard for each sampled architecture.

3.2 Overall Architecture

We propose an architecture-distribution segmentation network, NADS (Neural Architecture Distribution Sampling), that models ambiguity by sampling discrete architectural instances at inference time (Fig. 1). Unlike CVAE- or diffusion-based methods, which inject an explicit latent variable (sampling $z \sim p(z)$ and generating $\hat{y}$ via $p_\theta(y \mid x, z)$ or an iterative noise-to-mask process), NADS places stochasticity on the model structure. It samples an architecture $a \sim q_\psi(a \mid x)$ and predicts through a deterministic forward pass $\hat{y}(x; a) = \mathbb{I}[\sigma(f_\theta(x; a)) > 0.5]$, so diversity comes from architecture sampling rather than latent-noise injection.

3.2.1 Architecture Variable

We define a segmentation network with L searchable positions, each corresponding to one SearchableBlock inserted into the backbone (e.g., several blocks in the decoder). At each position $\ell \in \{1, \ldots, L\}$, the network provides a finite set of candidate operators, $\mathcal{O}_\ell = \{o_{\ell,1}, o_{\ell,2}, \ldots, o_{\ell,C_\ell}\}$, where each $o_{\ell,c}$ is a shape-preserving mapping from a feature map to another feature map with the same tensor shape. This ensures that all operator choices are compatible with the same backbone topology.

The architecture variable is a collection of discrete choices $a = (a_1, \ldots, a_L)$ with each $a_\ell \in \{1, \ldots, C_\ell\}$ selecting one candidate operator at position ℓ; sampling a thus instantiates a concrete architecture. The global network layout remains unchanged; only the internal operator choices at these L positions vary across samples.

3.2.2 ARCHITECTURE DISTRIBUTION

We learn a conditional probability distribution over architectures $q_\psi(a \mid x)$, parameterized by an architecture controller g_ψ that produces input-dependent logits from image features. Given an input image x, we extract a feature representation $h(x)$ from the backbone encoder, and the controller outputs $\{s_\ell(x)\}_{\ell=1}^{L}$, where $s_\ell(x) \in \mathbb{R}^{C_\ell}$ controls the selection probability over $\mathcal{O}_\ell$. We assume the conditional distribution factorizes across positions:

$$q_\psi(a \mid x) = \prod_{\ell=1}^{L} q_\psi(a_\ell \mid x), \quad q_\psi(a_\ell = c \mid x) = \left[\mathrm{softmax}(s_\ell(x))\right]_c. \tag{2}$$

Here, $s_\ell(x) = g_\psi(h(x))_\ell$. This factorization avoids enumerating an exponential number of full architectures, while retaining a large space for efficient sampling and learning. Diversity arises from composing different local operator choices across positions conditioned on the input. Although factorized, the induced conditional distribution over full architectures remains expressive because the number of possible architectures is $|\mathcal{A}| = \prod_{\ell=1}^{L} C_\ell$.

3.2.3 WEIGHT INHERITANCE

We denote the weight-inheritance network by f_θ: a sampled architecture a produces logits $z(x; a) = f_\theta(x; a)$ and probabilities $p(x; a) = \sigma(z(x; a))$. Randomness enters only through a; conditioned on a the forward pass is deterministic. All architectures inherit the same parameters θ but activate different operator (and hence parameter) subsets of the shared supernetwork.

3.2.4 TRAINING AND INFERENCE

At inference, stochastic segmentations are generated by sampling architectures $a^{(k)} \sim q_\psi(a \mid x)$ and thresholding, $\hat{y}^{(k)} = \mathbb{I}[\sigma(f_\theta(x; a^{(k)})) > 0.5]$ for $k = 1, \ldots, K$. The prediction set $\{\hat{y}^{(k)}\}_{k=1}^{K}$ approximates the model's conditional output distribution and supports evaluation with distributional metrics and downstream uncertainty estimation. Direct optimization of discrete architectural choices is not differentiable. During training, we use a differentiable relaxation to update ψ, while still training the inherited weights θ with sampled architectures. At each searchable position, we draw a relaxed one-hot vector $\tilde{a}_\ell = \mathrm{GumbelSoftmax}(s_\ell(x), \tau)$ using a straight-through Gumbel-Softmax distribution (Jang et al., 2017) with temperature τ. This relaxation enables gradient-based learning of ψ. In later stages, we reduce τ to encourage sharper distributions and more discrete behavior.

3.3 OBJECTIVE FUNCTION

Pixel-wise supervision against a single annotation can drive the model toward an "average" mask that matches no plausible annotator. To learn a distribution of valid masks, we employ set-level supervision: a set of sampled predictions is trained to match the set of available annotations for each image.

3.3.1 GED SURROGATE LOSS

For the k-th sampled architecture $a^{(k)} \sim q_\psi(a \mid x)$, we denote the predicted probability map as $p^{(k)}(x) \triangleq \hat{p}(x; a^{(k)}) = \sigma(f_\theta(x; a^{(k)}))$. We adopt sample-based energy distance with a soft-IoU ground metric. We use the soft-IoU distance

$$d_{\mathrm{IoU}}(u, v) \triangleq 1 - \frac{\langle u, v \rangle}{\|u\|_1 + \|v\|_1 - \langle u, v \rangle + \epsilon}, \tag{3}$$

where $u \in [0, 1]^{H \times W}$ and $v \in [0, 1]^{H \times W}$. Then we compute the following averages:

$$\bar{d}_{PY}(x) = \frac{1}{KM} \sum_{k=1}^{K} \sum_{m=1}^{M} d_{\mathrm{IoU}}\left(p^{(k)}(x), y^{(m)}(x)\right), \tag{4}$$

$$\bar{d}_{PP}(x) = \frac{1}{K(K-1)} \sum_{\substack{k,k'=1 \\ k \neq k'}}^{K} d_{\mathrm{IoU}}\left(p^{(k)}(x), p^{(k')}(x)\right). \tag{5}$$

Our set-level loss is an energy-distance surrogate. Relative to the classic GED, we drop the annotation-only term $\bar{d}_{YY}$ (constant w.r.t. model parameters), reweight the diversity term by β, and clip the result at zero to preserve non-negativity:

$$\mathcal{L}_{\text{set}}(x) = \max\Big(0,\ 2\,\bar{d}_{PY}(x) - \beta\,\bar{d}_{PP}(x)\Big), \tag{6}$$

where β controls the strength of the diversity term among predictions. The cross term $\bar{d}_{PY}$ encourages sampled predictions to be close to the annotation set, while the negative self term $-\beta\,\bar{d}_{PP}$ discourages mode collapse by penalizing overly similar predictions.

3.3.2 Full Objective

In addition, we add an auxiliary stabilizer acting on the mean prediction and the mean annotation to ease early optimization. Let $z^{(k)}(x) = f_\theta(x; a^{(k)})$ denote the logit map produced by the k-th sampled architecture, and define $\bar{z}(x) = \frac{1}{K}\sum_{k=1}^{K} z^{(k)}(x)$ and $\bar{y}(x) = \frac{1}{M}\sum_{m=1}^{M} y^{(m)}(x)$. We define $\mathcal{L}_{\text{aux}}(x)$ as a standard BCE–Dice loss applied to $\bar{z}(x)$ and $\bar{y}(x)$. It is weighted by a small λ_{aux}, keeping $\mathcal{L}_{\text{set}}$ as the dominant objective. Finally, we regularize the architecture distribution using negative entropy, $\mathcal{R}(q_\psi) = -\mathbb{E}_x \sum_{\ell=1}^{L} H(q_\psi(a_\ell \mid x))$, to avoid premature collapse. The full training objective over a batch with weights λ_{aux} and λ_{arch} is

$$\mathcal{L} = \mathbb{E}_x\Big[\mathcal{L}_{\text{set}}(x) + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}(x)\Big] + \lambda_{\text{arch}}\mathcal{R}(q_\psi). \tag{7}$$

3.4 Variable Architecture Search

Stochasticity comes from sampling among discrete candidate blocks at each searchable position, so we first construct a position-specific candidate bank by an efficient evolutionary procedure and then learn the architecture distribution over the retained candidates. Each candidate at position ℓ is a shape-preserving module ($o_{\ell,c} : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}$) specified by a compact genotype, so it can be swapped without changing tensor shapes or the global backbone topology. We use aging evolution with a fixed-size population. Each cycle selects a tournament's best individual as the parent, mutates it into a child, and evaluates the child under a fixed budgeted protocol with the same stochastic-segmentation objective. The child is then inserted and the oldest individual removed, which encourages exploration and reduces premature convergence. After the search, we retain the top-N candidates per position as the final bank. An individual's fitness combines distributional quality with a complexity proxy,

$$\text{fit}(g) = (1 - \text{HM-IoU}) + \lambda_{\text{ged}}\text{GED} + \lambda_{\text{cost}}\text{Cost}(g), \tag{8}$$

where $\text{Cost}(g)$ is a complexity proxy accumulated from the selected genes, biasing the search toward lower-cost active paths. The full algorithm and hyper-parameters are given in the supplementary material.

4 Experiments

4.1 Main Experiments

We evaluate on four settings: stochastic segmentation on LIDC-IDRI with multi-annotator supervision; additional stochastic benchmarks on Hippocampus and QUBIQ Prostate under the Tyche protocol (Rakic et al., 2024; Li et al., 2024); multi-class deterministic segmentation on Cityscapes (Cordts et al., 2016); and single-class curvilinear segmentation on Crack500, which has thin structures and severe class imbalance (Zhang et al., 2016). Unless otherwise stated, we follow the dataset protocols of (Li et al., 2025) and (Chen et al., 2024).

For deterministic datasets, the architecture distribution is not used for distribution matching; the controller minimizes the expected standard segmentation loss, converges to a near-deterministic policy, and uses maximum a posteriori (MAP) inference at test time. With a single annotation per image (M=1), we train the controller with the standard Dice loss.

Table 1: Quantitative results on LIDC-IDRI, with methods listed in chronological order. All baseline results are taken from (Li et al., 2025). Our results are over 3 seeds.

Method	GED_{16}	GED_{32}	GED_{50}	GED_{100}	HM-IoU_{16}	HM-IoU_{32}
Prob. U-Net (Kohl et al., 2018)	$0.310_{\pm 0.01}$	$0.303_{\pm 0.01}$	–	$0.252_{\pm 0.004}$	$0.552_{\pm 0.00}$	$0.548_{\pm 0.00}$
HProb. U-Net (Kohl et al., 2019)	$0.270_{\pm 0.01}$	–	–	–	$0.530_{\pm 0.01}$	–
PhiSeg (Baumgartner et al., 2019)	$0.262_{\pm 0.00}$	$0.247_{\pm 0.00}$	–	$0.224_{\pm 0.004}$	$0.586_{\pm 0.00}$	$0.595_{\pm 0.00}$
SSN (Monteiro et al., 2020)	$0.259_{\pm 0.00}$	$0.243_{\pm 0.01}$	–	$0.225_{\pm 0.002}$	$0.558_{\pm 0.00}$	$0.555_{\pm 0.01}$
cFlow (Selvan et al., 2020)	–	$0.225_{\pm 0.01}$	–	–	–	$0.584_{\pm 0.00}$
CAR (Kassapis et al., 2021)	–	–	–	$0.228_{\pm 0.009}$	–	–
JProb. U-Net (Zhang et al., 2022b)	–	$0.206_{\pm 0.01}$	–	–	–	$0.647_{\pm 0.01}$
PixelSeg (Zhang et al., 2022a)	$0.243_{\pm 0.01}$	–	–	–	$0.614_{\pm 0.01}$	–
MoSE (Gao et al., 2022)	$0.218_{\pm 0.03}$	$0.195_{\pm 0.002}$	$0.195_{\pm 0.002}$	$0.189_{\pm 0.002}$	$0.624_{\pm 0.004}$	–
AB (Chen et al., 2022)	$0.213_{\pm 0.01}$	$0.196_{\pm 0.02}$	$0.193_{\pm 0.002}$	–	$0.614_{\pm 0.01}$	$0.619_{\pm 0.001}$
CIMD (Rahman et al., 2023)	$0.234_{\pm 0.005}$	$0.218_{\pm 0.005}$	$0.210_{\pm 0.005}$	–	$0.587_{\pm 0.01}$	$0.592_{\pm 0.002}$
CCDM (Zbinden et al., 2023)	$0.212_{\pm 0.002}$	$0.194_{\pm 0.001}$	$0.187_{\pm 0.002}$	$0.183_{\pm 0.002}$	$0.623_{\pm 0.002}$	$0.631_{\pm 0.002}$
GMM_NF (Li et al., 2025)	$0.196_{\pm 0.001}$	$0.189_{\pm 0.001}$	$0.185_{\pm 0.002}$	$0.184_{\pm 0.000}$	$0.641_{\pm 0.002}$	$0.644_{\pm 0.001}$
Ours	$\mathbf{0.172}_{\pm 0.001}$	$\mathbf{0.169}_{\pm 0.001}$	$\mathbf{0.165}_{\pm 0.003}$	$\mathbf{0.164}_{\pm 0.003}$	$\mathbf{0.658}_{\pm 0.001}$	$\mathbf{0.671}_{\pm 0.002}$

Table 2: Additional stochastic-segmentation benchmarks beyond LIDC-IDRI under the Tyche evaluation protocol (Rakic et al., 2024) (context size 16 for in-context methods; prediction set of size 8): Hippocampus (brain MRI) and QUBIQ (Li et al., 2024) Prostate Task 1/2. We report Max Dice and GED (the generalized energy distance reported as $\text{GED}^2/\text{GED}_8$ in Tyche). Baselines marked $*$ are taken from Tyche; "–" marks a metric not reported for that method. NADS (Ours) is averaged over 3 seeds (std 0.01).

Method	Hippocampus		QUBIQ Prostate T1		QUBIQ Prostate T2	
	Max Dice	GED	Max Dice	GED	Max Dice	GED
UniverSeg* (Butoi et al., 2023)	$0.84_{\pm 0.01}$	–	$0.91_{\pm 0.01}$	–	$0.88_{\pm 0.03}$	–
SegGPT* (Wang et al., 2023)	$0.10_{\pm 0.01}$	–	$0.94_{\pm 0.01}$	–	$0.89_{\pm 0.03}$	–
SENet* (Roy et al., 2020)	$0.68_{\pm 0.01}$	–	$0.83_{\pm 0.02}$	–	$0.83_{\pm 0.03}$	–
SAM* (Kirillov et al., 2023)	$0.71_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.90_{\pm 0.01}$	$0.20_{\pm 0.03}$	$0.85_{\pm 0.03}$	$0.31_{\pm 0.06}$
SAM-Med2d* (Cheng et al., 2023)	$0.52_{\pm 0.01}$	$0.93_{\pm 0.02}$	$0.62_{\pm 0.04}$	$0.80_{\pm 0.09}$	$0.64_{\pm 0.06}$	$0.78_{\pm 0.11}$
Tyche-IS* (Rakic et al., 2024)	$0.87_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.12_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.20_{\pm 0.05}$
Tyche-TS* (Rakic et al., 2024)	$0.88_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.95_{\pm 0.01}$	$0.09_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.15_{\pm 0.03}$
PhiSeg* (Baumgartner et al., 2019)	$0.88_{\pm 0.00}$	$0.14_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.17_{\pm 0.05}$
Prob. U-Net* (Kohl et al., 2018)	$0.91_{\pm 0.00}$	$0.13_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.91_{\pm 0.03}$	$0.18_{\pm 0.05}$
CIMD* (Rahman et al., 2023)	$0.84_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.87_{\pm 0.02}$	$0.26_{\pm 0.04}$
CCDM (Zbinden et al., 2023)	–	–	$0.94_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.14_{\pm 0.01}$
Ours	$\mathbf{0.95}_{\pm 0.01}$	$\mathbf{0.09}_{\pm 0.01}$	$\mathbf{0.96}_{\pm 0.01}$	$\mathbf{0.06}_{\pm 0.01}$	$\mathbf{0.94}_{\pm 0.01}$	$\mathbf{0.10}_{\pm 0.01}$

4.1.1 Evaluation Metrics

For LIDC we report GED_K and HM-IoU_K between the K-sample prediction set and the annotation set under multiple K, reflecting distribution quality as the number of samples increases. For Hippocampus and QUBIQ Prostate we follow the Tyche protocol (Max Dice, GED); for Cityscapes we report IoU and for Crack500 Precision, Recall, and F1. Lower GED is better; higher HM-IoU, Max Dice, IoU, Precision, Recall, and F1 are better. The supplementary material provides expanded algorithmic and hyper-parameter details for reproducibility.

4.1.2 Stochastic Segmentation

Table 1 reports the quantitative results on LIDC, with methods listed in chronological order; all baseline values are taken from (Li et al., 2025). Our approach obtains the lowest GED under every sampling budget and the highest HM-IoU among the compared methods. Our method attains $\text{HM-IoU}_{16} = 0.658$ and $\text{HM-IoU}_{32} = 0.671$, exceeding the previously reported baseline with the highest HM-IoU on each metric ($\text{HM-IoU}_{16} = 0.641$ by GMM_NF and $\text{HM-IoU}_{32} = 0.647$ by JProb. U-Net) by $+1.7$ and $+2.4$ points, respectively. For GED, our method reduces the lowest baseline values of 0.196, 0.189, 0.185, and 0.183 to 0.172, 0.169, 0.165, and 0.164 for $K = \{16, 32, 50, 100\}$. Since GED penalizes both missing annotation modes and samples that are far from the annotator set, these gains indicate that architecture sampling increases diversity while keeping the samples aligned with the empirical multi-rater distribution. Qualitative examples in the supplementary material show that the sampled predictions produce multiple plausible masks, varying in size and shape in a way that mirrors annotator differences rather than collapsing to a single averaged contour.

Table 3: Disentangling the stochastic objective, sampler, candidate-bank construction, and complexity-aware search on LIDC with $K = 16$.

Sampler / model	Bank / objective	HM-IoU	GED	Best-Dice	$\bar{d}_{PP}$
Fixed architecture	Evolved bank + GED	0.536	0.981	0.762	0.041
Random architecture sampler	Evolved bank + GED	0.598	0.341	0.801	0.126
Input-agnostic $q(a)$	Evolved bank + GED	0.628	0.233	0.816	0.148
Input-conditioned $q_\psi(a \mid x)$	Random bank + GED	0.639	0.215	0.822	0.153
Input-conditioned $q_\psi(a \mid x)$	Evolved bank, no cost + GED	0.651	0.181	0.828	0.162
Input-conditioned $q_\psi(a \mid x)$	Evolved bank + best-candidate Dice	0.612	0.327	**0.833**	0.098
Input-conditioned $q_\psi(a \mid x)$	Evolved bank + Hungarian IoU	0.641	0.284	0.819	0.121
16-network ensemble	Independent weights + GED	0.623	0.276	0.817	0.134
Input-conditioned $q_\psi(a \mid x)$	Evolved bank + GED	**0.658**	**0.172**	0.831	0.167

4.1.3 Additional Stochastic Benchmarks

We further evaluate three multi-annotation settings from the Tyche protocol (Rakic et al., 2024): Hippocampus (brain MRI) and QUBIQ (Li et al., 2024) Prostate Task 1/2, all with a prediction set of size 8 (Table 2). Across all three settings, NADS attains the highest Max Dice and lowest GED: on Hippocampus it improves GED $0.13 \rightarrow 0.09$ and Max Dice $0.91 \rightarrow 0.95$, and on QUBIQ Prostate reaches GED $0.06/0.10$ and Max Dice $0.96/0.94$ (Task 1/2), improving over baselines such as Prob. U-Net and Tyche-TS. Thus architecture sampling transfers beyond LIDC-IDRI to different organs and annotation protocols, rather than being specific to a single dataset.

4.1.4 Architectural Provenance

Each hypothesis is produced by one sampled architecture and is described by its discrete code $a = (a_1, \ldots, a_L)$ (L=12 integers, ≈ 28 bits); since the forward pass is deterministic given a and the shared weights θ, the code reproduces its mask. Unlike a continuous latent or a denoising trajectory, this code has positional structure: *where* an entry sits in the decoder indicates *what* it controls. We show this by single-position intervention, fixing all positions at the MAP code, sweeping one position's operator over its alternatives, and measuring the resulting change of the output mask. Aggregated by decoder stage, the controlled attribute varies with depth: coarse stages mainly change overall size/extent (mean ΔArea 66–72%, vs. 45% at the finest stage), while finer stages govern boundary sharpness (ΔBoundary $0.12 \rightarrow 0.25$) and connected-component structure (Δ#Comp $0.04 \rightarrow 0.23$). Every single-position edit moves the output (mean ΔIoU 0.23–0.35), and the full stage-wise table and per-position map are in the supplementary material. The 28-bit code is therefore a structured, position-interpretable provenance trace, and each code position ties to a nameable mask property, mirroring the distinct ambiguity types (extent, weak-boundary inclusion, connectivity) that motivate our approach.

4.1.5 Deterministic Segmentation Transfer

Although primarily motivated by stochastic segmentation, NADS also applies to deterministic settings; here the controller is trained to minimize the expected standard segmentation loss and MAP inference is used at test time. On Cityscapes (multi-class urban scenes, baselines from (Li et al., 2025)), the activated MAP subnet (26M parameters) reaches 74.2 IoU, above the highest baseline IoU under the same protocol. On Crack500 (thin curvilinear cracks with severe class imbalance, baselines from (Chen et al., 2024)), NADS achieves 71.02 Precision, 70.53 Recall, and 70.97 F1, improving over prior approaches including CrackFormer (Liu et al., 2021a) and JTFN+CIRL (Chen et al., 2024). These results show the architecture-distribution framework is not restricted to multi-annotator medical ambiguity. The full per-baseline tables and qualitative comparisons are given in the supplementary material.

4.2 Inference Efficiency

For a fixed input, many stochastic segmentation models cache the encoder features across hypotheses, so per-sample cost is dominated by a lightweight component (latent head, decoder, logit/prototype sampler, flow, or denoising step) rather than the full backbone. NADS follows the same reusable-feature principle, but its repeated component is an architecture-dependent active path: features are computed once, architectures are sampled from $q_\psi(a \mid x)$, and each hypothesis executes only its selected path, giving $O(E + KD_a)$ per sample for a per-sample active-path cost D_a. The

activated subnet averages 26M parameters while the full supernet stores 387M, so deployment cost is determined by the sampled path rather than the stored bank. Under a feature-cached protocol, NADS attains the lowest latency at $K{=}1$ and $K{=}16$ (28.6 ms and 94.8 ms) among representative stochastic baselines (Prob. U-Net (Kohl et al., 2018), PHiSeg (Baumgartner et al., 2019), SSN (Monteiro et al., 2020), Pancakes (Rakic et al., 2025), cFlow (Selvan et al., 2020), GMM_NF (Li et al., 2025), CCDM (Zbinden et al., 2023)), and is faster than Pancakes at all tested budgets. The full per-method latency table and the offline training/search-cost table, both with all baselines, are given in the supplementary material.

4.3 Ablation Studies

We conduct ablation studies to clarify why both architecture sampling and set-level supervision are necessary for architecture-distribution stochastic segmentation. Unless otherwise stated, ablations are evaluated on LIDC using sampled prediction sets and report HM-IoU, GED, and coverage-oriented metrics.

4.3.1 Architecture Sampler and Set-Level Objective

We disentangle the contributions of the architecture sampler, candidate-bank construction, complexity-aware search, and stochastic objective (Table 3, $K{=}16$). Fixed architecture + GED scores lowest, showing the set-level loss alone cannot create a good stochastic distribution without a sampling mechanism. Random and input-agnostic samplers improve, and an input-conditioned sampler over an evolved, complexity-aware bank is best; the random-bank and no-cost-search rows further isolate the effect of the evolved support from the learned sampler. On the objective, best-candidate Dice raises its own Best-Dice metric but lowers HM-IoU/GED (best-of-K leaves many sampled architectures weakly constrained), and Hungarian matching trails GED because only matched prediction–annotation pairs receive direct supervision. A 16-network ensemble with independent weights uses more trained parameters than our shared supernet, yet still scores below NADS (0.623 vs 0.658 HM-IoU). The gain therefore comes from architecture sampling over a shared, evolved bank rather than raw model capacity, and each hypothesis executes only a 26M active path.

4.3.2 Standard Losses and Gradient-Free Selection

We next replace the set-level GED surrogate with conventional pixel-wise losses (BCE, Dice), following the common practice of randomly selecting one annotation as the target at each iteration. This produces a consistent failure mode: the model *loses stochasticity* as the architecture distribution collapses to a near-deterministic configuration, with one operator dominating most positions. Distributional metrics degrade (Dice/BCE reach only HM-IoU$_{16}{=}0.49/0.44$ and GED$_{16}{=}0.89/0.85$; full per-variant numbers in the supplementary) because single-label supervision optimizes a point estimate with no incentive to match the multi-modal set. We also learn the selection logits without gradients, treating them as an evolvable genotype updated by aging evolution on validation fitness. This variant is likewise unstable and collapses to near-deterministic selections (HM-IoU$_{16}{=}0.57$, GED$_{16}{\approx}1.0$): the fitness is noisy and non-stationary as θ evolves, so black-box evolution over-exploits early winners. These results motivate our gradient-based learning of the controller ψ jointly with θ.

5 Conclusion

We introduced architecture distributions as a stochastic source for segmentation: diversity is generated by sampling a discrete architecture $a \sim q_\psi(a \mid x)$ and producing each hypothesis through its active path. Unlike latent-code or denoising-noise sources, this support is a discrete architecture space that can be shaped by evolutionary search before distribution learning, and set-level supervision avoids averaging collapse while retaining architectural provenance. On LIDC-IDRI our model obtains the lowest GED$_{16}$ (0.172) and highest HM-IoU$_{16}$ (0.658), with inference executing only sampled active paths. Future work can enlarge and optimize this support toward distribution matching and calibrated uncertainty.

AI USE STATEMENT

In this work, we used generative AI tools to polish wording, shorten sentences, and check grammar in text that the authors had already drafted. We have reviewed all AI-assisted text and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses only publicly available, de-identified benchmark datasets and involves no human subjects, and we are not aware of ethical concerns raised by it. We note that the medical segmentation results reported here are retrospective benchmark results and are not validated for clinical use.

REPRODUCIBILITY STATEMENT

The information needed to reproduce our results is given in the main text and the appendix. The architecture parameterization, the set-level energy-distance objective, and the evolutionary construction of the candidate bank are specified in Section 3. The appendix details the searchable-block design and candidate genotypes (Appendix A.1), the supernet warm-up protocol (Appendix A.2), the full aging-evolution procedure in pseudo-code (Appendix A.3), and the complete hyper-parameter settings (Appendix A.4). Dataset splits and evaluation protocols follow the references cited in Section 4, and the extended ablations report the per-variant numbers summarized in the main text.

REFERENCES

Lalith Bharadwaj Baru, Kamalaker Dadi, Tapabrata Chakraborti, and Raju S. Bapi. Ambiguous medical image segmentation using diffusion schrödinger bridge. In James C. Gee, Daniel C. Alexander, Jaesung Hong, Juan Eugenio Iglesias, Carole H. Sudre, Archana Venkataraman, Polina Golland, Jong Hyo Kim, and Jinah Park (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, pp. 25–35, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-04965-0.

Christian F. Baumgartner, Kerem C. Tezcan, Krishna Chaitanya, Andreas M. Hötker, Urs J. Muehlematter, Khoschy Schawkat, Anton S. Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 119–127, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-32244-1. doi: 10.1007/978-3-030-32245-8_14. URL https://doi.org/10.1007/978-3-030-32245-8_14.

Ishaan Bhat, Josien P. W. Pluim, and Hugo J. Kuijf. Generalized probabilistic u-net for medical image segementation. In Carole H. Sudre, Christian F. Baumgartner, Adrian Dalca, Chen Qin, Ryutaro Tanno, Koen Van Leemput, and William M. Wells III (eds.), *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging*, pp. 113–124, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-16749-2.

Victor Ion Butoi, Jose Javier Gonzalez Ortiz, Tianyu Ma, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. UniverSeg: Universal Medical Image Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/pdf/1908.09791.pdf.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Proceedings of the European Conference on Computer Vision Workshops(ECCVW)*, 2022.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Zhuangzhuang Chen, Zhuonan Lai, Jie Chen, and Jianqiang Li. Mind marginal non-crack regions: Clustering-inspired representation learning for crack segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12698–12708, June 2024.

Junlong Cheng, Jin Ye, Zhongying Deng, et al. SAM-Med2D, 2023.

Mingfei Cheng, Kaili Zhao, Xuhong Guo, Yajing Xu, and Jun Guo. Joint topology-preserving and feature-refinement network for curvilinear structure segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7127–7136, 2021. doi: 10.1109/ICCV48922.2021.00706.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Fabio De Sousa Ribeiro, Omar Todd, Charles Jones, Avinash Kori, Raghav Mehta, and Ben Glocker. Flow stochastic segmentation networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14754–14765, October 2025.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Zhitong Gao, Yucong Chen, Chuyu Zhang, and Xuming He. Modeling multimodal aleatoric uncertainty in segmentation with mixture of stochastic experts. *arXiv preprint arXiv:2212.07328*, 2022.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in neural information processing systems*, 32, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Elias Kassapis, Georgi Dikov, Deepak K Gupta, and Cedric Nugteren. Calibrated adversarial refinement for stochastic semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7057–7067, 2021.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Simon A. A. Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus H. Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6965–6975, Montréal, Canada, 2018. Curran Associates Inc.

Simon A. A. Kohl, Bernardino Romera-Paredes, Klaus H. Maier-Hein, Danilo Jimenez Rezende, S. M. Ali Eslami, Pushmeet Kohli, Andrew Zisserman, and Olaf Ronneberger. A hierarchical probabilistic u-net for modeling multi-scale ambiguities. *CoRR*, abs/1905.13077, 2019. URL http://arxiv.org/abs/1905.13077.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Conghui Li, Chern Hong Lim, and Xin Wang. GMM-based VAE model with normalising flow for effective stochastic segmentation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=MxYvh8zpbR.

Hongwei Bran Li, Fernando Navarro, Ivan Ezhov, Amirhossein Bayat, Dhritiman Das, Florian Kofler, Suprosanna Shit, Diana Waldmannstetter, Johannes C. Paetzold, Xiaobin Hu, Benedikt Wiestler, Lucas Zimmer, Tamaz Amiranashvili, Chinmay Prabhakar, Christoph Berger, Jonas Weidner, Michelle Alonso-Basant, Arif Rashid, Ujjwal Baid, Wesam Adel, Deniz Ali, Bhakti Baheti, Yingbin Bai, Ishaan Bhatt, Sabri Can Cetindag, Wenting Chen, Li Cheng, Prasad Dutand, Lara Dular, Mustafa A. Elattar, Ming Feng, Shengbo Gao, Henkjan Huisman, Weifeng Hu, Shubham Innani, Wei Jiat, Davood Karimi, Hugo J. Kuijf, Jin Tae Kwak, Hoang Long Le, Xiang Lia, Huiyan Lin, Tongliang Liu, Jun Ma, Kai Ma, Ting Ma, Ilkay Oksuz, Robbie Holland, Arlindo L. Oliveira, Jimut Bahan Pal, Xuan Pei, Maoying Qiao, Anindo Saha, Raghavendra Selvan, Linlin Shen, Joao Lourenco Silva, Ziga Spiclin, Sanjay Talbar, Dadong Wang, Wei Wang, Xiong Wang, Yin Wang, Ruiling Xia, Kele Xu, Yanwu Yan, Mert Yergin, Shuang Yu, Lingxi Zeng, YingLin Zhang, Jiachen Zhao, Yefeng Zheng, Martin Zukovec, Richard Do, Anton Becker, Amber Simpson, Ender Konukoglu, Andras Jakab, Spyridon Bakas, Leo Joskowicz, and Bjoern Menze. Qubiq: Uncertainty quantification for biomedical image segmentation challenge, 2024. URL https://arxiv.org/abs/2405.18435.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3763–3772, 2021a. doi: 10.1109/ICCV48922.2021.00376.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b.

Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: modelling spatially correlated aleatoric uncertainty. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Agata Mosinska, Pablo Marquez-Neila, Mateusz Kozinski, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3136–3145, 2018. doi: 10.1109/CVPR.2018.00331.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4095–4104. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/pham18a.html.

Aimon Rahman, Jeya Maria Jose Valanarasu, Ilker Hacihaliloglu, and Vishal M. Patel. Ambiguous medical image segmentation using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11536–11546, June 2023.

Marianne Rakic, Hallee E. Wong, Jose Javier Gonzalez Ortiz, Beth A. Cimini, John V. Guttag, and Adrian V. Dalca. Tyche: Stochastic in-context learning for medical image segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11159–11173, 2024. doi: 10.1109/CVPR52733.2024.01061.

Marianne Rakic, Siyu Gai, Etienne Chollet, John V. Guttag, and Adrian V. Dalca. Pancakes: Consistent multi-protocol image segmentation across biomedical domains, 2025. URL https://arxiv.org/abs/2512.13534.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33014780. URL https://doi.org/10.1609/aaai.v33i01.33014780.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 'Squeeze & Excite' Guided Few-Shot Segmentation of Volumetric Images. *Medical Image Analysis*, 59:101587, 2020.

Raghavendra Selvan, Frederik Faye, Jon Middleton, and Akshay Pai. Uncertainty quantification in medical image segmentation with normalizing flows. In Mingxia Liu, Pingkun Yan, Chunfeng Lian, and Xiaohuan Cao (eds.), *Machine Learning in Medical Imaging*, pp. 80–90, Cham, 2020. ISBN 978-3-030-59861-7.

M. M. Amaan Valiuddin, Christiaan G. A. Viviers, Ruud J. G. van Sloun, Peter H. N. de With, and Fons van der Sommen. Improving aleatoric uncertainty quantification in multi-annotated medical image segmentation with normalizing flows. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*, pp. 75–88, Cham, 2021. Springer International Publishing. ISBN 978-3-030-87735-4.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43 (10):3349–3364, 2020.

Wei Wang, Kaicheng Yu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Recurrent u-net for resource-constrained segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2142–2151, 2019. doi: 10.1109/ICCV.2019.00223.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. SegGPT: Segmenting Everything in Context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rylqooRqK7.

Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris C. Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for uncertainty estimation and dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Lukas Zbinden, Lars Doorenbos, Theodoros Pissas, Adrian Thomas Huber, Raphael Sznitman, and Pablo Márquez-Neila. Stochastic segmentation with conditional categorical diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1119–1129, 2023. doi: 10.1109/ICCV51070.2023.00109.

Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, 2016. doi: 10.1109/ICIP.2016.7533052.

Wei Zhang, Xiaohong Zhang, Sheng Huang, Yuting Lu, and Kun Wang. Pixelseg: Pixel-by-pixel stochastic semantic segmentation for ambiguous medical images. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 4742–4750, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548060. URL https://doi.org/10.1145/3503161.3548060.

Wei Zhang, Xiaohong Zhang, Sheng Huang, Yuting Lu, and Kun Wang. A probabilistic model for controlling diversity and accuracy of ambiguous medical image segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 4751–4759, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548115. URL https://doi.org/10.1145/3503161.3548115.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00889-5.

SUPPLEMENTARY MATERIAL

A IMPLEMENTATION DETAILS

A.1 NETWORK ARCHITECTURE

We insert shape-preserving SearchableBlocks into the decoder feature stream. Concretely, after each decoder upsampling stage, we place n_{search} searchable blocks, resulting in $L = 4 \times n_{\text{search}}$ searchable positions in total. Each SearchableBlock keeps the input/output tensor shape identical ($\mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}$), so it can be dropped into the decoder without changing skip connections or resolution transitions.

Each searchable position ℓ maintains a finite candidate set $\mathcal{O}_\ell = \{o_{\ell,1}, \ldots, o_{\ell,C_\ell}\}$. Candidates are compiled from a compact genotype describing micro-architecture choices, including block family (residual/MBConv/ConvNeXt/identity), kernel size, dilation, expansion ratio, number of layers, optional SE, activation, DropPath, and LayerScale. Unless stated otherwise, we use the same default pool size across positions, and later keep a smaller Top-N subset after evolution.

A.2 SUPERNET WARMUP

Warmup is used to obtain reliable inherited weights for the large operator bank before evolutionary selection and architecture-distribution learning. In practice, warmup trains only the shared operator weights while disabling all architecture-parameter updates: we freeze the architecture-related parameters and optimize only the supernet weights with AdamW. At each iteration, we create a set of K_{train} predictions for the same mini-batch by repeating:

1. uniformly sample one operator at every searchable position;
2. execute a single-path forward pass through the instantiated subnetwork.

This produces K_{train} hypotheses $\{\hat{y}^{(k)}\}_{k=1}^{K_{\text{train}}}$ stacked as $(B, K_{\text{train}}, 1, H, W)$, which are optimized with the same set-level stochastic segmentation loss used later in training. Uniform per-position sampling during warmup ensures that all candidates receive sufficient gradient updates, so the bank becomes broadly functional rather than overfitting to a small subset of operators.

After warmup, we unfreeze architecture parameters and proceed to evolutionary candidate selection and end-to-end training with the reduced Top-N banks.

A.3 EVOLUTIONARY ALGORITHM

The evolutionary procedure is used to construct a performance-oriented, position-specific candidate bank for each searchable position. Concretely, we start from a warm-started supernet (Phase 1), run aging evolution on a held-out search-validation split, and finally keep Top-N candidates per position to form the bank used in end-to-end training (Phase 3). The detailed procedure is given in Algorithm 1.

A.3.1 SEARCH SPACE

We define L searchable positions in the decoder. At each position ℓ, a candidate operator is a shape-preserving module ($\mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}$), so it can be swapped without changing the global U-Net topology. Each operator is specified by a compact genotype

$$g = (\texttt{block_type}, k, d, e, n, r, \texttt{act}, p_{\texttt{dp}}, \gamma_{\texttt{ls}}), \tag{9}$$

where $\texttt{block_type} \in$ {resconv, mbconv, convnext, identity}, $k \in \{3, 5, 7\}$ is the kernel size, $d \in \{1, 2\}$ is the dilation, $e \in \{2, 3, 4, 6\}$ is the expansion ratio, $n \in \{1, 2, 3\}$ is the number of internal layers (with constraints for some block types), $r \in \{0, 0.125, 0.25\}$ is the SE ratio, $\texttt{act} \in$ {relu, silu, gelu} is the activation, p_{dp} is the DropPath rate, and γ_{ls} is the LayerScale initialization. The full architecture is represented as a length-L index vector $a = [a_1, \ldots, a_L]$, where a_ℓ selects one candidate from the local pool at position ℓ.

Algorithm 1 Aging evolution

Require: Warm-started supernet f_θ, candidate pools $\{\mathcal{O}_\ell\}_{\ell=1}^{L}$, search-validation loader $\mathcal{D}_{sv}$, and the hyperparameters listed in Sec. A.3.
Ensure: Top-N banks $\{\mathcal{B}_\ell\}_{\ell=1}^{L}$.
Initialize population $\mathcal{P} \leftarrow \emptyset$, history $\mathcal{H} \leftarrow \emptyset$.
for $i = 1$ to P **do**
  Sample a random genotype index vector $a^{(i)}$ (biased against identity).
  Evaluate $\mathrm{fit}(a^{(i)})$ on $\mathcal{D}_{sv}$ with the budgeted protocol.
  Insert $a^{(i)}$ into $\mathcal{P}$ and $\mathcal{H}$.
end for
for $t = 1$ to T **do**
  Sample a subset $\mathcal{S} \subset \mathcal{P}$ with $|\mathcal{S}| = S$.
  Parent $a^p \leftarrow \arg\min_{a \in \mathcal{S}} \mathrm{fit}(a)$.
  Child $a^c \leftarrow \mathrm{Mutate}(a^p, p_{\text{mut}})$ (change 1–3 positions).
  Evaluate $\mathrm{fit}(a^c)$ on $\mathcal{D}_{sv}$ (lock genotype; optional local fine-tune; metric evaluation; cost proxy).
  Insert a^c into $\mathcal{P}$ and $\mathcal{H}$.
  if $|\mathcal{P}| > P$ **then**
    Remove the oldest individual from $\mathcal{P}$ (aging)
  end if
end for
Select elites $\mathcal{E} \leftarrow$ the best E individuals from $\mathcal{H}$ by fitness.
for $\ell = 1$ to L **do**
  Rank candidate indices at position ℓ by mean fitness within $\mathcal{E}$.
  Set $\mathcal{B}_\ell \leftarrow$ Top-N candidates (with deterministic fill if needed).
end for
return $\{\mathcal{B}_\ell\}_{\ell=1}^{L}$.

A.3.2 Initialization

Before evolution, we build an initial pool of C candidate genotypes per position by random sampling with a fixed random seed, while enforcing uniqueness via a hash key of the genotype fields. In our experiments, we set $C = 10$ candidates per position. This pool is created once at warmup time and then reused during evolution and final training.

A.3.3 Aging evolution

We use regularized aging evolution with a fixed population size. At each cycle, we sample a small subset tournament, select the best individual as parent, mutate it to produce a child, evaluate the child with a budgeted protocol, then insert the child into the population and remove the oldest individual to keep population size constant. This aging mechanism encourages exploration and reduces premature convergence.

Given a parent index vector a, we mutate it with rate p_{mut} by randomly changing 1 to 3 positions; for each chosen position, the current index is replaced with another valid index from the corresponding position pool, excluding the current one. This mutation has low computational cost and keeps the global topology unchanged.

A.3.4 Fitness evaluation

To evaluate an individual a, we lock the supernet to the genotype specified by a, instantiate the corresponding single-path subnet, and compute its HM-IoU, GED, and complexity-aware fitness on the search-validation loader under a fixed evaluation budget. For each input, we sample architectures from the induced distribution over the restricted candidate sets and execute a single-path forward pass to obtain a prediction set. We generate K_{eval} sampled predictions per input to interface with the same set-based evaluation code, and we limit evaluation to at most B_{eval} images to control cost.

Furthermore, we perform a short local fine-tune on the search-validation stream to improve genotype distinguishability. We fine-tune only the shared weights for t_{ft} steps with the locked genotype, then

restore the original shared weights from a snapshot to avoid contaminating the supernet. Architecture parameters remain frozen throughout fine-tuning. During this local fine-tune, we keep the genotype locked in the same way and sample architectures as above, while updating only the shared weights. We combine distributional metrics with a complexity proxy:

$$\mathrm{fit}(a) \;=\; (1 - \text{HM-IoU}) \;+\; \lambda_{\mathrm{ged}}\mathrm{GED} \;+\; \lambda_{\mathrm{cost}}\mathrm{Cost}(a), \tag{10}$$

where $\mathrm{Cost}(a)$ accumulates per-position genotype complexity scores:

$$\begin{aligned} \mathrm{Cost}(a) &= \sum_{\ell=1}^{L} \mathrm{cost}(g_{\ell,a_\ell}), \\ \mathrm{cost}(g) &\propto (k^2)(C^2)\max(n,1)\cdot\eta(g). \end{aligned} \tag{11}$$

and $\eta(g)$ applies small multiplicative factors for MBConv/ConvNeXt expansion, dilation, and SE usage. Lower fitness is better.

A.3.5 TOP-N EXTRACTION.

After evolution, we take the best E elites from the entire history ranked by fitness. For each position, we compute the mean fitness of each candidate index among elites, rank candidates by this mean, and keep the Top-N genotypes. If fewer than N unique indices appear among elites at a position, we fill the bank with unused pool candidates and with additional randomly sampled non-identity genotypes. The resulting Top-N per-position banks are then used to instantiate the final training model.

A.4 PARAMETER SETTINGS

This section summarizes the default hyperparameters used in our experiments, following the released implementation.

A.4.1 DATASET SETTING

LIDC-IDRI is a thoracic CT dataset for lung cancer screening/diagnosis, where lesions are annotated by multiple radiologists, making it suitable for studying annotation ambiguity. We partition the dataset into three disjoint subsets: train for supernet warmup and final training, search-val used only for evolutionary fitness evaluation and training-time monitoring, and a test set sealed until the final report. Concretely, we set search_val_split=0.10 and test_split=0.20, and only apply paired geometric augmentation on the training subset (random flips and 90° rotations).

Hippocampus is included as an additional stochastic segmentation benchmark to evaluate whether NADS transfers beyond LIDC-IDRI. We follow the Tyche evaluation protocol (Rakic et al., 2024): baseline numbers are taken from Tyche, the context size is 16 for in-context methods, the prediction-set size is 8, and the reported metrics are GED^2 and Max Dice. Our NADS result is evaluated under the same prediction-set size and metric protocol.

Crack500 is a pixel-wise pavement crack segmentation dataset. Following the standard protocol, the dataset is provided as 500 high-resolution images and further processed into 3,368 cropped samples of size 360×640. We use the official split with 1,897 training images, 347 validation images, and 1,124 testing images. During training, we apply horizontal flipping, random cropping, and random rotations of $90^\circ/180^\circ/270^\circ$; additionally, all training samples are cropped to 256×256 for optimization, and all methods are trained and evaluated at 256×256 resolution.

Cityscapes provides fine-grained pixel annotations for urban driving scenes with 19 semantic classes, serving as a representative multi-class deterministic segmentation benchmark. We use the official split containing 2,975 training images and 500 validation images at native resolution 512×1024. Following the referenced setting, we use a resized resolution of 256×512 for both training and evaluation.

A.4.2 PHASE 1: WARMUP

We warm up the shared supernet weights using SPOS-style single-path uniform sampling: at each step, each searchable position randomly selects one operator, and we aggregate K such stochastic forward passes to form a prediction set for set-level supervision. During warmup, architecture parameters ψ are frozen and only inherited weights θ are updated with AdamW. We use warmup_steps=500, warmup_K=4, and warmup_lr=2×10^{-4}.

A.4.3 PHASE 2: AGING EVOLUTION

Unless otherwise stated, we use a pool size per position of $C = 10$ with a retained bank size per position of $N = 5$; a population size of $P = 40$, tournament size of $S = 10$, $T = 120$ cycles, and mutation probability $p_{\text{mut}} = 0.30$; $E = 30$ retained elites for Top-N extraction; evaluation settings of $K_{\text{eval}} = 16$ forward passes per input and $B_{\text{eval}} = 120$ images; local fine-tuning for $t_{\text{ft}} = 30$ steps with learning rate 5×10^{-5} and $K_{\text{ft}} = 4$ passes per step; and fitness weights $\lambda_{\text{ged}} = 1.0$ and $\lambda_{\text{cost}} = 0.02$.

A.4.4 PHASE 3: TRAINING

We train the final NADS model for 200 epochs with batch_size=8 and num_workers=4. We use two AdamW optimizers: one for inherited weights θ (lr_theta=1×10^{-4}, weight_decay=1×10^{-4}) and one for architecture parameters ψ (lr_psi=3×10^{-4}, weight_decay=0). We freeze ψ for the first warmup_epochs=$0.1 \times$ `epochs` and then jointly optimize (θ, ψ). For the training objective, we use a diversity weight $\beta = 1.0$ in $\mathcal{L}_{\text{set}}$, a small auxiliary-loss weight $\lambda_{\text{aux}} = 0.1$, and an architecture-regularizer weight $\lambda_{\text{arch}} = 0.01$ that is annealed toward 0 together with the temperature τ.

We employ a straight-through Gumbel-Softmax relaxation with temperature τ during training and progressively anneal it to encourage more discrete behavior. Concretely, ψ is frozen for the first $0.1E$ epochs (with E the total number of epochs), after which τ is linearly annealed from 5.0 to 0.5:

$$\tau(e) = \begin{cases} 5.0, & e < 0.1E, \\ \max\left(0.5,\ 5.0 - 4.5\,\dfrac{e - 0.1E}{0.9E}\right), & e \geq 0.1E, \end{cases}$$

where e is the current epoch. The entropy regularizer coefficient and the uniform-mix probability are also decayed over training. For stochastic prediction, we sample $K_{\text{pred}} = 16$ architectures per input (threshold at 0.5, consistent with Eq. (6)) and evaluate with $\tau_{\text{eval}} = 0.7$; we cap GED/HM-IoU evaluation at `max_images_for_ged`=200 images during monitoring.

B CAPABILITY EVALUATION

We ablate the number of searchable positions (L) and the per-position retained operator budget (Top-N) to understand the capacity–performance trade-off under a shorter 100-epoch setting (Table 4). Each setting is trained and evaluated three times, and we report the average. As shown in Table 4, increasing L from 4 to 12 generally improves both accuracy and distributional fidelity under a moderate K (with Top-5, HM-IoU increases from 0.613 to 0.622, while GED decreases from 0.241 to 0.231). This indicates that more positions provide additional stage-wise flexibility for input-conditioned operator selection and better alignment with multi-annotation uncertainty. In contrast, using an overly large retained set (Top-10) can be suboptimal, likely because low-ranked operators add optimization noise and weaken the quality of the learned operator distribution, leading to degraded HM-IoU/GED in some settings.

The method also remains effective at small capacity. With L=4 and Top-3 per position, corresponding to 12 available operator choices in total, the model achieves HM-IoU 0.606 and GED 0.261. This suggests that a large operator pool is not required for competitive distributional quality. The observation is consistent with the intuition that, when the retained per-position budget N is at least the number of dominant annotation modes, the model can allocate enough distinct operators to cover the major modes; we treat this as an empirical observation rather than a guarantee. Overall, L=12 with Top-5 yields the best trade-off among the evaluated configurations and is used as the default configuration.

Table 4: Ablation on the number of searchable positions and the retained candidate bank size (Top-N per position).

#Positions	Top-3		Top-5		Top-10	
	HM-IoU	GED	HM-IoU	GED	HM-IoU	GED
4	0.606	0.261	0.613	0.241	0.608	0.245
8	0.609	0.241	0.617	0.237	0.600	0.264
12	0.611	0.241	**0.622**	**0.231**	0.608	0.255

Table 5: Ablation of the Gumbel-Softmax temperature τ and annealing strategy under straight-through sampling. We report the average gate entropy $\mathcal{H}$ across all searchable positions. $\mathcal{H}_{0.1T}$ and $\mathcal{H}_T$ denote the average gate entropy measured at 10% and 100% of training, respectively.

Setting	τ schedule	$\mathcal{H}_{0.1T}$	$\mathcal{H}_T$	HM-IoU	GED
Fixed high τ	$\tau = 5.0$	1.46	1.38	0.615	0.242
Fixed low τ	$\tau = 0.5$	0.62	0.25	0.607	0.252
Piecewise anneal	$5.0 \rightarrow 0.5$	1.44	0.53	**0.630**	**0.233**

C TEMPERATURE τ EVALUATION

Our architecture controller parameterizes an input-conditioned categorical distribution $q_\psi(a \mid x) = \prod_{\ell=1}^{L} q_\psi(a_\ell \mid x)$ over discrete operator choices, from which we sample architectures to generate K hypotheses at inference time. Since directly optimizing discrete architectural choices is non-differentiable, during training we adopt the straight-through Gumbel-Softmax relaxation with temperature τ (Eq. (7) in the main paper) to obtain a relaxed one-hot vector $\tilde{a}_\ell$. As described in the main paper, we progressively reduce τ in later stages to encourage sharper distributions and more discrete behavior, better matching the discrete sampling used at inference.

We compare three strategies during training: Fixed-High temperature, Fixed-Low temperature, and Annealed temperature (start high and progressively decrease to a low value). We report distributional quality metrics together with an optimization diagnostic, the average architecture entropy $H(q_\psi(a_\ell \mid x))$. All experiments are run for 100 epochs.

All results are summarized in Table 5. Fixed-High keeps the relaxation overly soft, so the controller learns under a "mixture-like" proxy while inference relies on discrete architecture sampling. This train–test mismatch typically hurts both segmentation quality and set-level matching. Fixed-Low makes the relaxation near-discrete from the beginning; the straight-through estimator becomes high-variance and the controller tends to prematurely concentrate on a few operators, leading to reduced entropy, even with the negative-entropy regularizer $\mathcal{R}(q_\psi)$. In contrast, the annealed schedule provides a stable exploration–exploitation trade-off: early training benefits from smoother gradients and exploration (higher entropy), while later training becomes more discrete to align with inference-time sampling, yielding the best distributional metrics.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 QUALITATIVE RESULTS ON LIDC

Fig. 2 shows qualitative LIDC examples. Columns (a)–(d) are ground-truth annotations from four experts, column (e) is the variance map over 16 predictions, and columns (f)–(i) are four architecture-sampled predictions from NADS. The sampled predictions vary in size and shape in a way that mirrors annotator differences, rather than collapsing to a single averaged contour.

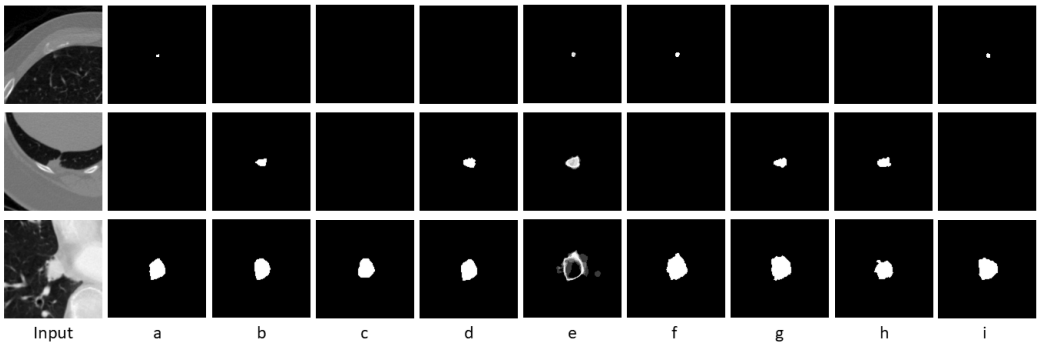


Figure 2: Qualitative results on LIDC. Columns (a)–(d): four expert annotations; column (e): variance map over 16 predictions; columns (f)–(i): four sampled predictions from NADS.

Table 6: Single-position code intervention grouped by decoder stage (MAP base; n=37 LIDC test images with non-empty MAP masks). Coarse stages predominantly control size; finer stages control boundary sharpness and connectivity.

Intervened stage	**ΔArea (%)**	**ΔBoundary**	**Δ#Comp**	**ΔIoU**
Stage-1 (pos 0–2, coarse)	**66.7**	0.121	0.036	0.234
Stage-2 (pos 3–5)	**72.1**	0.120	0.045	0.306
Stage-3 (pos 6–8)	54.1	0.164	0.099	0.351
Stage-4 (pos 9–11, fine)	45.4	**0.250**	**0.234**	0.334

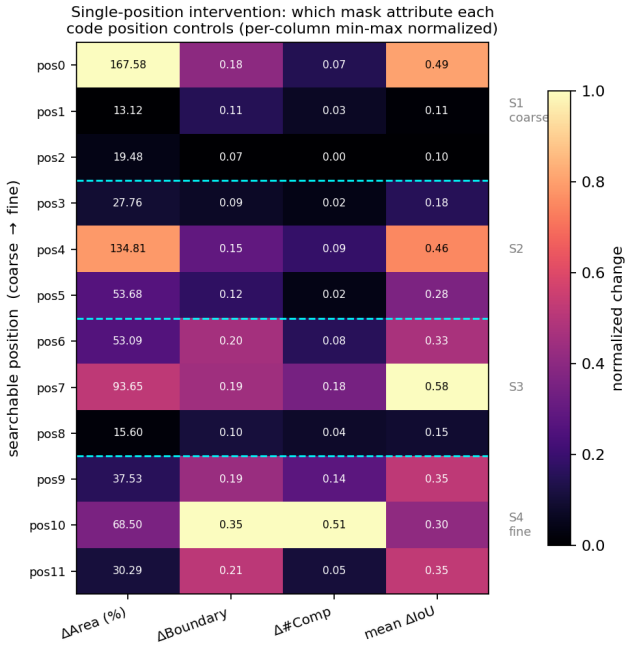


Figure 3: Per-position single-position-intervention map.

D.2 Architectural Provenance

The main paper reports the stage-wise single-position intervention findings inline. Table 6 gives the full stage-wise statistics and Fig. 3 the per-position map. Fixing all positions at the MAP code and sweeping one operator over its alternatives, we report the mean change of each mask attribute; mean ΔIoU confirms a single-position edit moves the output. Each column of the map is min-max normalized, with raw values annotated. Intervening at coarse positions mainly changes overall mask area, while finer positions increasingly change boundary sharpness and connected components.

Table 7: Quantitative results on Cityscapes. Baselines are taken from (Li et al., 2025). "Param" is the executed MAP subnet size for NADS; the full supernet stores 387M parameters.

Method	Backbone	Param	IoU
DeepLabv3 (Chen et al., 2017)	ResNet50	39M	58.6
DeepLabv3 (Chen et al., 2017)	ResNet101	58M	59.2
UPerNet (Xiao et al., 2018)	ResNet101	83M	60.7
HRNet (Wang et al., 2020)	w48v2	70M	63.3
UPerNet (Liu et al., 2021b)	Swin-Tiny	58M	65.5
CCDM (Zbinden et al., 2023)	-	30M	60.3
CCDM (Zbinden et al., 2023)	Dino ViT-S	50M	65.8
Prob. U-Net (Kohl et al., 2018)	-	33M	63.2
GMM_NF (Li et al., 2025)	-	38M	73.0
Ours	-	26M	**74.2**

Table 8: Quantitative results on Crack500. Baselines are taken from (Chen et al., 2024).

Method	Precision	Recall	F1
UNet (Ronneberger et al., 2015)	62.22	68.85	61.83
VGG-UNet (Mosinska et al., 2018)	58.18	60.26	51.79
TopoNet (Hu et al., 2019)	66.81	62.68	60.06
DRU (Wang et al., 2019)	61.94	**71.43**	62.82
CrackFormer (Liu et al., 2021a)	69.13	66.24	64.75
JTFN (Cheng et al., 2021)	68.81	69.06	65.76
JTFN+CIRL (Chen et al., 2024)	70.32	69.93	67.62
Prob. U-Net (Kohl et al., 2018)	56.80	69.39	60.80
Ours	**71.02**	70.53	**70.97**

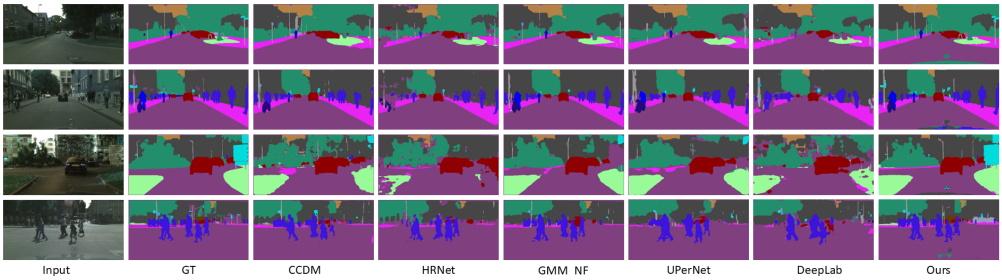


Figure 4: Qualitative results on Cityscapes. Following (Li et al., 2025), all baselines are trained and evaluated at a resolution of 256×512.

D.3 Deterministic Segmentation Transfer

Although NADS is primarily motivated by stochastic segmentation, we also evaluate deterministic transfer on Cityscapes and Crack500. For deterministic datasets, the architecture distribution is not used for distribution matching; the controller is trained to minimize the expected standard segmentation loss and MAP inference is used at test time.

Table 7 summarizes the Cityscapes results. NADS reaches 74.2 IoU, above the strongest baseline under the same evaluation protocol. Qualitative examples are shown in Fig. 4; the predictions show cleaner region consistency and better separated object–background boundaries than the compared outputs.

Table 8 reports results on Crack500, where the goal is to segment thin crack structures. NADS achieves 71.02 Precision, 70.53 Recall, and 70.97 F1, improving over prior approaches including CrackFormer (Liu et al., 2021a) and JTFN+CIRL (Chen et al., 2024). Qualitative examples are shown in Fig. 5.

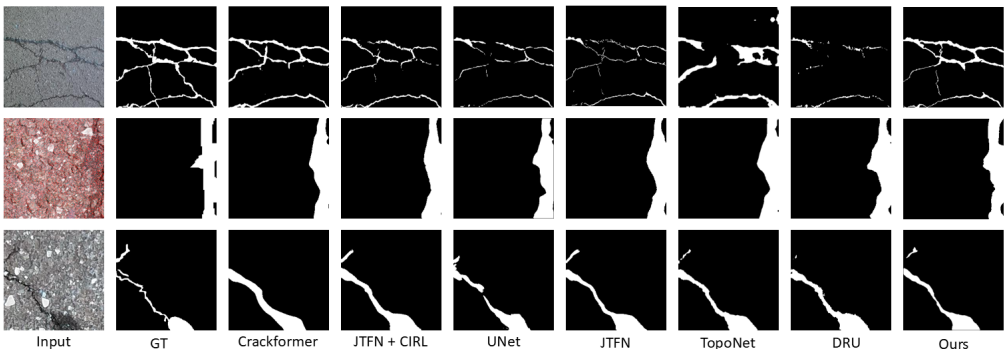


Figure 5: Qualitative results on Crack500. Following (Chen et al., 2024), all baselines are trained and evaluated at a resolution of 256×256.

Table 9: Model complexity and inference efficiency. Active Params reports the parameters executed by one sampled forward path (averaged over sampled subnets for Ours). Inference time is measured under a feature-cached protocol for generating K samples from one input at fixed resolution on the same hardware, where deterministic image features are computed once and reused when applicable. Each setting is repeated 10 times and reported as mean and standard deviation.

Method	Active Params	Sampling Complexity	Inference Time$_{K=1}$	Inference Time$_{K=16}$	Inference Time$_{K=32}$
Prob. U-Net (Kohl et al., 2018)	33M	$O(E+KH)$	$34.58_{\pm 0.07}$	$98.4_{\pm 0.6}$	$\mathbf{166.3}_{\pm 0.9}$
PHiSeg (Baumgartner et al., 2019)	41M	$O(E+KD)$	$42.10_{\pm 0.08}$	$104.7_{\pm 0.8}$	$172.5_{\pm 1.1}$
SSN (Monteiro et al., 2020)	34M	$O(E+KS_{\text{logit}})$	$42.8_{\pm 0.3}$	$100.6_{\pm 1.0}$	$169.4_{\pm 1.5}$
Pancakes (Rakic et al., 2025)	45M	$O(E+KS_{\text{prot}})$	$37.5_{\pm 0.4}$	$116.8_{\pm 1.3}$	$208.6_{\pm 2.2}$
cFlow (Selvan et al., 2020)	34M	$O(E+KL_{\text{flow}})$	$139.84_{\pm 0.11}$	$635.2_{\pm 1.8}$	$1163.7_{\pm 2.6}$
GMM_NF (Li et al., 2025)	38M	$O(E+KL_{\text{flow}})$	$213.02_{\pm 0.14}$	$1026.4_{\pm 2.4}$	$1893.8_{\pm 3.5}$
CCDM (Zbinden et al., 2023)	30M	$O(E+KT)$	$1322.64_{\pm 5.83}$	$19480.6_{\pm 9.7}$	$38110.2_{\pm 17.5}$
Ours	26M(AVG)	$O(E+KD_a)$	$\mathbf{28.6}_{\pm 4.2}$	$\mathbf{94.8}_{\pm 9.5}$	$171.9_{\pm 18.7}$

Table 10: Offline training/search cost and peak memory. Full Params reports stored parameters during training, while Active Params reports the parameters executed by one sampled path.

Method	Full Params	Active Params	Peak VRAM	Time/Epoch	Search + Train Time
Prob. U-Net (Kohl et al., 2018)	33M	33M	669.6 MB	2:28	8.2 h
PHiSeg (Baumgartner et al., 2019)	41M	41M	823.3 MB	3:13	10.7 h
SSN (Monteiro et al., 2020)	34M	34M	910.0 MB	3:05	10.3 h
Pancakes (Rakic et al., 2025)	45M	45M	1180.0 MB	3:30	11.7 h
GMM_NF (Li et al., 2025)	38M	38M	738.7 MB	5:52	19.6 h
CCDM (Zbinden et al., 2023)	30M	30M	1460.0 MB	2:35	8.6 h
Ours	387M	26M(AVG)	2005.0 MB	9:42	35.0 h

D.4 Inference Efficiency and Offline Cost

We compare inference efficiency with representative stochastic segmentation baselines in Table 9. The cost can be decomposed into a deterministic image-feature term E and a sample-specific stochastic term. For a fixed input, many stochastic segmentation models can cache encoder features across multiple hypotheses, so increasing K mainly repeats the latent head, decoder, logit/prototype sampler, flow, or denoising component rather than the full backbone. NADS follows the same reusable-feature principle, but the repeated component is an architecture-dependent active path. In a cached implementation, encoder/controller features are computed once, architectures are sampled from $q_\psi(a \mid x)$, and each hypothesis executes only its selected path.

Table 10 separates deployment cost from offline cost. NADS is more expensive during training because it maintains the full candidate bank, warms inherited weights, runs candidate-bank evolution, and then trains the input-conditioned controller. This cost is paid before deployment. For a new domain, the candidate bank can be searched once because useful operators can depend on image statistics and annotation ambiguity; for related domains, the same bank can be reused as an initialization while fine-tuning the controller and inherited weights.

D.5 Extended Ablation Studies

The main paper's disentangling ablation—covering the architecture sampler, candidate-bank construction, complexity-aware search, and the set-level objective (fixed/random/input-agnostic/input-

Table 11: Ablations on LIDC ($K=16, 32$). Standard single-label losses and evolution-only selection both cause the architecture distribution to collapse toward a near-deterministic network, leading to poor distributional metrics.

Variant	$K=16$		$K=32$	
	HM-IoU	**GED**	**HM-IoU**	**GED**
Dice	0.492	0.895	0.496	0.882
BCE	0.437	0.854	0.448	0.849
Evolution selection	0.572	1.000	0.573	0.998
Ours	0.658	0.172	0.671	0.169

conditioned samplers, evolved vs. random bank, GED vs. best-candidate Dice vs. Hungarian, and a 16-network ensemble)—is reported in the main body. Here we detail the standard-loss and gradient-free-selection ablations that did not fit in the main text.

Table 11 reports the standard-loss and evolution-only ablations. With BCE or Dice supervision, the model tends to lose stochasticity because single-label supervision optimizes a point-estimation objective and provides no explicit incentive to maintain sample diversity. Learning the architecture-selection distribution without gradients is also unstable: black-box evolution over-exploits early winners under noisy, non-stationary validation fitness and collapses toward near-deterministic selections.