
GauSAM: Contour-Guided 2D Gaussian Fields for Multi-Scale Medical Image Segmentation with Segment Anything

Jinxuan Wu^{1*} Jiange Wang^{1*} Dongdong Zhang^{1†}

¹School of Computer Science and Technology

Tongji University

{2154146, 2252699, ddzhang}@tongji.edu.cn

Abstract

Effective multiscale medical image segmentation requires simultaneously preserving smooth spatial continuity and accurately delineating high-frequency boundaries, yet pixel-wise decoders often fail to maintain this balance consistently across varying resolutions. We introduce GauSAM, which seamlessly integrates contour-guided 2D Gaussian probability fields into the Segment Anything Model to address these challenges. In our framework, segmentation masks are parameterized as continuous probability fields of learnable 2D Gaussian primitives, enforcing spatially smooth and structurally consistent. Contourlet transforms extract rich multidirectional frequency information, notably edges and fine textures, which dynamically guide the spatial distribution of Gaussian primitives to substantially improve boundary fidelity in complex structures. The incorporation of these high-frequency contour priors also enriches the expressive capacity of the SAM image encoder. Extensive experiments on diverse 2D medical segmentation tasks confirm that GauSAM consistently delivers robust generalization and state-of-the-art performance with only 1.2M trainable parameters. The official implementation of GauSAM is publicly available at <https://github.com/Quinten-Wu504/GauSAM>.

1 Introduction

Medical image segmentation demands high precision and smooth boundary delineation, as anatomical structures often exhibit ambiguous edges and segmentation errors can lead to serious clinical consequences [1]. To address this, a wide range of methods have been proposed, including classical architectures such as U-Net [2] and several fine-tuned variants of the Segment Anything Model (SAM) [3]. These approaches typically rely on fixed-resolution, grid-based predictions and discrete interpolation schemes, which introduce discretization artifacts and limit their accuracy, especially in high-resolution or cross-scale scenarios [4, 5, 6].

To overcome the limitations of discrete segmentation, implicit neural representations (INRs) [7] have emerged as a resolution-agnostic alternative, mapping continuous spatial coordinates to feature representations. However, existing INRs still struggle to capture fine-grained structures due to their random sampling strategies, which often neglect local contours and directional textures. Furthermore, their reliance on discrete latent codes can obscure complex signals, ultimately reducing their ability to produce sharp and coherent segmentation outputs.

Motivated by the limitations of both discrete and INR-based methods, we propose to formulate segmentation as a continuous Gaussian probability field. We are the first to model SAM’s segmentation

*These authors contributed equally to this work.

†Corresponding author.

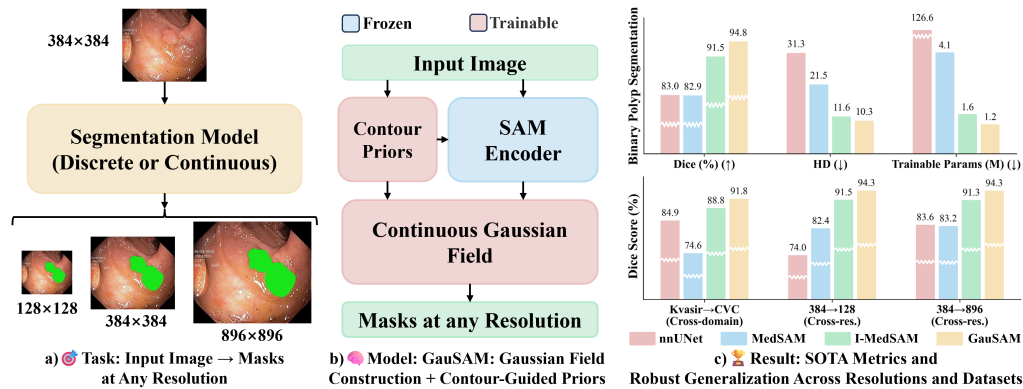


Figure 1: **Overview of GauSAM.** a) Illustration of resolution-agnostic mask generation: GauSAM processes a single image and produce segmentation masks at user-specified resolutions b) Schematic of GauSAM’s architecture and workflow: an input image is encoded by a frozen SAM backbone, enhanced with contour priors derived from the Non-Subsampled Contourlet Transform (NSCT), and processed through a trainable continuous Gaussian field to generate resolution-agnostic segmentation masks. c) Summary of results: GauSAM achieves state-of-the-art Dice and Hausdorff Distance across in-domain, cross-domain, and cross-resolution settings, outperforming baselines.

probability field as a continuous representation parameterized by 2D Gaussian Splatting [8] for mask generation, enabling resolution-agnostic predictions while mitigating grid-related artifacts. Unlike prior works that employ Gaussian Splatting merely as auxiliary priors or post-processing components [9, 10, 11, 12], our framework is trained end-to-end, allowing the Gaussian parameters to be directly optimized with task supervision. However, standard Gaussian kernels lack directional sensitivity, often resulting in blurry or misaligned contours in the final mask result [13].

To address this issue, we propose Contour-Guided Gaussian-Field SAM (GauSAM) — a novel framework that enhances the SAM encoder with multiscale contour priors and constructs a contour-aware Gaussian field decoder. Specifically, GauSAM leverages the Non-Subsampled Contourlet Transform (NSCT) [14] to extract rich boundary features, which are progressively fused into the encoder via a Contour Propagation Unit (CPU). On the decoder side, we construct a resolution-agnostic probability field from a set of learnable Gaussian primitives, whose parameters are modulated by the contour-enhanced encoded features. A contour-guided spatial adjustment module further aligns these primitives to semantic boundaries, yielding high-fidelity masks with sharp contours and strong structural coherence.

Our main contributions are as follows:

1. We are the first to represent SAM’s segmentation probability field as a continuous Gaussian probability field implemented via learnable 2D Gaussian Splatting, instantiated from a parameterized template library via Gumbel-Softmax sampling and modulated by contour-enhanced semantic features for stable training and compact representation.
2. We integrate NSCT-based structural priors into the SAM encoder via a lightweight Contour Propagation Unit, enhancing multiscale and multidirectional sensitivity to fine contours.
3. We propose a Contour-Guided Position Drift module to predict the spatial offsets of each Gaussian primitive from NSCT contour cues, improving boundary alignment.
4. GauSAM achieves superior performance over state-of-the-art discrete and INR-based methods across diverse 2D medical segmentation benchmarks, with notable robustness in cross-domain and resolution-agnostic inference.

2 Related Work

2D Medical Image Segmentation Since the introduction of pixel-level segmentation by FCN [15], deep convolutional architectures such as U-Net [2] and its massive variant [16, 17, 18, 19]

have consistently achieved SOTA performance across a wide range of medical imaging modalities. More recently, the Vision Transformer–based Segment Anything Model (SAM) [3] has demonstrated remarkable zero-shot segmentation capabilities. To adapt SAM to the medical imaging domain, researchers have proposed various lightweight fine-tuning methods [20, 21, 22], all of which substantially improve the SAM encoder’s ability to perceive medically relevant structures and features. However, these existing methods rely on fixed-resolution representations, which are prone to introducing discretization artifacts and blurred boundaries, especially in high-resolution or cross-scale scenarios [23]. To address these limitations, implicit neural representation (INR) methods [7, 24, 25] employ continuous neural fields to yield subpixel-smooth outputs at arbitrary resolutions, yet they still struggle with preserving precise texture structures and lacking interpretability [26].

Contourlet-Based Enhancement The Contourlet transform achieves a multiscale, multidirectional subband decomposition via a cascade of a Laplacian pyramid and directional filter banks, efficiently capturing curves at arbitrary angles and high-frequency contour information [27]. Compared to traditional wavelets, Contourlets offer superior curve representation and directional selectivity, leading to widespread use in image and feature enhancement [27]. Leveraging these same properties, the non-subsampled Contourlet transform (NSCT) [14, 28, 29], which eschews downsampling via non-subsampled Laplacian pyramids and filter banks, achieves shift invariance and superior directional sensitivity compared to the conventional Contourlet transform. Recently proposed hybrid architectures that fuse Contourlet subband features with deep learning models [30, 31] have significantly enhanced the accuracy and robustness of fine-structure delineation, further validating the complementary role of frequency-domain enhancement in complex medical imaging tasks.

Gaussian Splatting for Continuous Representation Gaussian Splatting (GS) explicitly represents a scene as a set of optimizable anisotropic Gaussian fields and leverages an efficient splatting-based rendering pipeline [8, 32]. In recent research, GS has shown broad potential across various vision tasks. In super-resolution reconstruction, GS-based approaches construct learnable 2D/3D Gaussian kernel radiance fields to enable real-time, high-fidelity interpolation at arbitrary upscaling factors [33, 34, 35, 36]. And in semantic segmentation, emerging studies reveal that GS can reconstruct feature and radiance fields with superior precision — without per-pixel network inference — while preserving spatial and semantic consistency, thus offering a promising paradigm for efficient and accurate semantic segmentation [37, 38, 39]. Motivated by these advances, our work is the first to employ GS to construct continuous probability fields that map high-dimensional features to segmentation masks, preserving sharp boundaries and semantic consistency across scales.

3 Methodology

3.1 Preliminaries

Let $X \in \mathbb{R}^{H \times W \times C}$ denote the input image defined on its native discrete pixel lattice, where H , W , and C denote the height, width, and number of channels respectively, and let $\Omega = [-1, 1]^2$ be the normalized continuous coordinate domain. We represent the segmentation target as a continuous mask probability field:

$$M : \Omega \rightarrow \Delta^{Class-1}, \quad \Delta^{Class-1} = \left\{ y \in \mathbb{R}^{Class} : y_c \geq 0 \forall c, \sum_{c=1}^{Class} y_c = 1 \right\} \quad (1)$$

where $Class$ is the number of semantic classes, and $\Delta^{Class-1}$ is the $(Class-1)$ -simplex of class probabilities. The segmentation task is thus formulated as modeling a parametric mapping:

$$\mathcal{T}_\Theta : X \mapsto M. \quad (2)$$

By sampling the continuous field M on an arbitrary target grid $p \subset \Omega$ of resolution $H' \times W'$, we obtain a discrete prediction $\hat{O} \in \mathbb{R}^{H' \times W' \times C}$ that can be compared to ground-truth annotations using standard differentiable loss functions. Such a continuous representation of the probability field inherently supports arbitrary output resolutions and preserves spatial coherence.

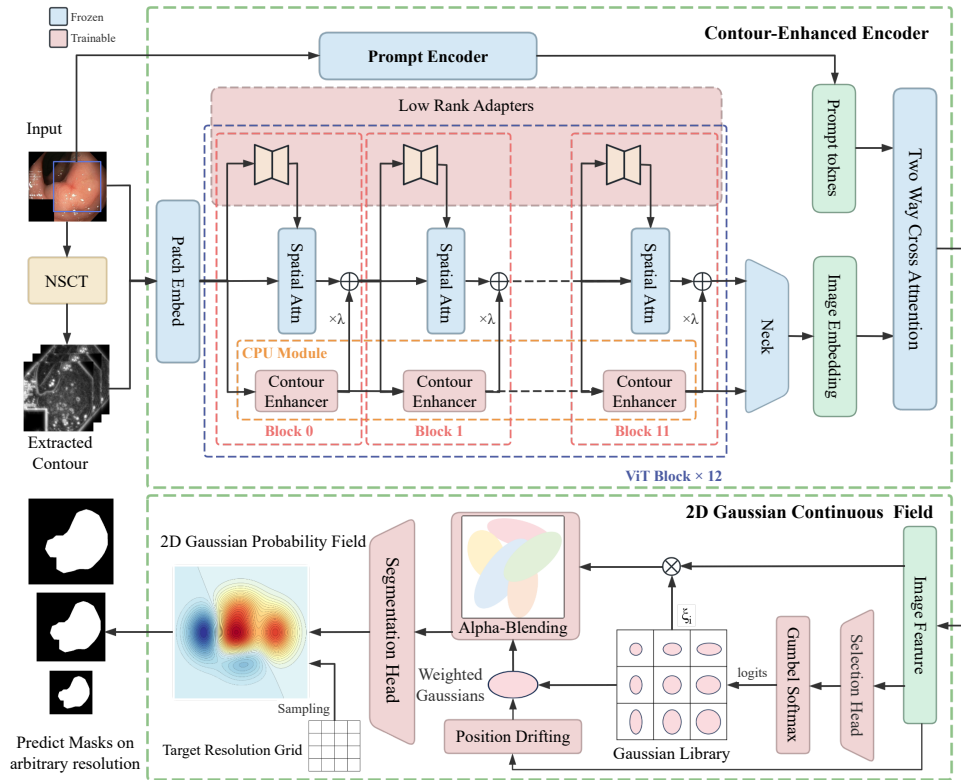


Figure 2: The overall architecture of GauSAM, consisting of Contour-Enhanced Encoding and 2D Gaussian Continuous Field Decoding. The input image is processed by a SAM image encoder, which is augmented by NSCT-based contour extraction and Contour Propagation Units (CPU). A continuous Gaussian field built from learnable 2D primitives then produces a resolution-agnostic segmentation mask. Red-colored modules indicate trainable components, including LoRA-adapted layers and newly proposed modules; blue modules denote frozen parts from the SAM-ViT-B backbone.

3.2 Overall Pipeline

As illustrated in Figure 2, GauSAM consists of two principal stages: (1) *Contour-Enhanced Image Encoding* and (2) *Continuous Field Decoding*.

In the first stage, the original SAM image encoder is enhanced with structural priors derived from a multi-level Non-Subsampled Contourlet Transform (NSCT) [14]. The resulting contour energy map E is progressively injected into each transformer block via a lightweight Contour Propagation Unit (CPU), which recursively refines E through adaptive modulation and soft fusion with semantic embeddings throughout the encoding hierarchy. In addition, the bounding-box prompt is encoded by SAM's prompt encoder as a sparse input, producing prompt embeddings that guide mask decoding through prompt-conditioned activations.

In the second stage, GauSAM constructs a continuous Gaussian probability field with weighted aggregation of N learnable 2D anisotropic Gaussian primitives, each modulated by opacity-scaled encoder features. This approach forms a continuous representation $\mathcal{X}(p)$, which is decoded into a resolution-agnostic probability mask via a lightweight segmentation head. Moreover, each Gaussian primitive selects its covariance and opacity from a learned template library through a Gumbel-Softmax-based mechanism. A contour-Guided Position Drift module is also introduced to adjust the location of each primitive, using contour-aware offsets and enhancing boundary alignment without compromising semantic coherence.

We provide further details on the contour-guided encoder in Section 3.3, and on the Gaussian primitive field construction and decoding process in Section 3.4.

3.3 Contour-Enhanced Image Encoder

Structural Contour Feature Extraction To enable high-fidelity modeling of fine-grained structures, we integrate into the image encoder a Structural Edge Extractor that leverages the Non-Subsampled Contourlet Transform (NSCT) to extract multiscale and multidirectional high-frequency contour features.

Given an input image $I \in \mathbb{R}^{B \times C \times H \times W}$, where B denotes the batch size, we apply an NSCT decomposition with L levels and a directional configuration $\Theta = \{\theta_1, \dots, \theta_n\}$ at each level:

$$\{ContourSubband_{l,\theta}\}_{l=1,\theta=1}^{L,\Theta_l} = NSCT(I) \quad (3)$$

To construct a unified structural prior without significantly increasing computational overhead, we aggregate all directional subband responses into a single contour-enhanced energy map E for each channel. We apply a point-wise nonlinear transformation to each directional coefficient $ContourSubband_{l,\theta}$, combining logarithmic compression and gamma correction. The aggregated map is formulated as:

$$E = \sum_{l=1}^L \sum_{\theta=1}^{\Theta_l} \log(1 + |ContourSubband_{l,\theta}|)^\gamma, \quad \gamma \in (0, 1) \quad (4)$$

The log space combination of coefficient maps, together with the gamma correction, reduces the dynamic range of contour features to suppress highlights, such as strong reflections caused by bodily fluids in rectal polyp images, thereby mitigating the impact of spurious strong contours on subsequent processing. The final structural map E is subsequently normalized to emphasize weak contours, thus retaining essential structural signals across scales and orientations with minimal computational complexity.

Contour Propagation Unit (CPU) To inject the extracted contour information into the ViT encoder in a progressive and semantically coherent manner, we design a novel Contour Propagation Unit (CPU). Instead of applying edge priors statically or at a single layer, CPU recursively propagates and refines contour information across the transformer hierarchy, ensuring that structural awareness is maintained through all encoding depths.

Each transformer block in the encoder receives two parallel inputs: the image embeddings F_ℓ and the contour feature embeddings E_ℓ , which are derived from NSCT, where ℓ indexes the ℓ -th transformer block from shallow to deep. Within each CPU-enhanced block, the contour signal is adaptively modulated through a lightweight MLP-based Edge Reweighting Module ψ :

$$R_\ell = \psi(\text{LN}(E_\ell)), \quad E'_\ell = R_\ell \odot E_\ell \quad (5)$$

Here, $\text{LN}(\cdot)$ denotes Layer Normalization, R_ℓ is a per-channel reweighting vector. The reweighted contour features are then softly fused into the image-embedding stream with a scaling parameter λ . Crucially, E'_ℓ is propagated to the next ViT block as $E_{\ell+1}$. This design introduces a structural inductive signal that complements the semantic representations in F_ℓ , improving boundary fidelity across layers.

3.4 2D Gaussian Continuous Probability Field

To enable the construction of a continuous probability field, we construct a 2D continuous Gaussian feature field that inherently bridges the gap between discrete encoder outputs and continuous probabilistic representations. By stacking elliptical kernels in a continuous field, Gaussian Splatting enforces spatially smooth and locally consistent representations. This mathematically grounded mechanism aligns with the inherent spatial coherence of semantic segmentation, wherein adjacent pixels are likely to belong to the same semantic entity. Concretely, the feature field is formulated as the weighted superposition of N learnable Gaussian primitives. Each 2D Gaussian primitive is modeled as an anisotropic Gaussian distribution:

$$G_i(p \mid \mu_i, \Sigma_i) = \frac{1}{2\pi |\Sigma_i|} \exp\left(-\frac{1}{2} (p - \mu_i)^\top \Sigma_i^{-1} (p - \mu_i)\right), \quad p \in \Omega \quad (6)$$

where i indexes the i -th Gaussian kernel, $\mu_i \in \Omega$ and $\Sigma_i \in \mathbb{R}^{2 \times 2}$ denote the center and a positive-definite covariance of a primitive, respectively. In situations where multiple Gaussian primitives overlap, we modulate each discrete encoder feature $v_i \in \mathbb{R}^k$ by its learned opacity $\xi_i \in \mathbb{R}$ via $f_i = \sigma(\xi_i) v_i$, where $\sigma(\cdot)$ denotes the sigmoid function that maps ξ_i into the interval $(0, 1)$, serving as a soft opacity for each basis function. The final continuous feature field is then defined as the weighted aggregation of all Gaussian contributions:

$$\mathcal{X}(p) = \sum_{i=1}^N G_i(p \mid \mu_i, \Sigma_i) \cdot f_i \quad (7)$$

Subsequently, the blended field $\mathcal{X}(p)$ is forwarded through a lightweight segmentation head $g : \mathbb{R}^D \rightarrow \mathbb{R}^C$, producing class logits $g(\mathcal{X}(p))$. These logits are further normalized via a point-wise softmax operation to yield the continuous probability mask $M(p)$. By sampling $M(p)$ over arbitrary spatial grid, we obtain the final segmentation mask in a resolution-agnostic manner.

Gaussian Parameters Library To address the optimization inefficiency in the above continuous feature field, associated with directly learning the covariance matrices Σ_i and opacity scalars ξ_i of the Gaussian primitives in an end-to-end approach, we introduce a learned library of m template pairs: $\{(\Sigma^{(k)}, \xi^{(k)})\}_{k=1}^m$. Each primitive then derives its parameters by selecting from this library in a differentiable manner. Specifically, for the i -th Gaussian primitive located at position p_i and associated with encoder feature v_i , a lightweight convolutional selection head projects v_i into an m -dimensional logit vector:

$$z_i = \text{selHead}(v_i) \in \mathbb{R}^m. \quad (8)$$

To enable discrete selection while retaining differentiability, we apply the Gumbel-Softmax operator to z_i , producing a convex weight vector:

$$w_i = \text{GumbelSoftmax}(z_i) \in \Delta^{m-1} \quad (9)$$

which defines a probability distribution over the library items. The final parameters Σ_i and ξ_i for the i -th Gaussian primitive are then obtained as weighted combinations of the template items:

$$\Sigma_i = \sum_{k=1}^m w_{i,k} \Sigma^{(k)}, \quad \xi_i = \sum_{k=1}^m w_{i,k} \xi^{(k)} \quad (10)$$

This parameterization constrains scale, anisotropy, and opacity of each primitive to lie within a compact, pre-learned subspace of the full Gaussian parameter space, improving training stability while preserving the expressive richness required for detailed Gaussian-splatting representations [33].

Contour-Guided Position Drift Existing Gaussian field approaches, owing to their uniformly distributed kernels, often struggle to capture boundary details adequately. We therefore propose a *Contour-Guided Position Drift* mechanism to enhance the contour representation in the Gaussian feature field. The core idea is to bias the placement of each Gaussian primitive toward semantic edges, thereby sharpening mask delineation without sacrificing spatial coherence. Specifically, for each initial pixel position p_i , we extract its contour feature vector $u_i \in \mathbb{R}^D$ from the *Contour-Guided Encoder* (Section 3.3), which contains adequate and multidirectional contour information. A lightweight MLP then predicts a small offset:

$$\Delta\mu_i = \tanh(\text{MLP}_{\text{off}}(u_i)) \quad (11)$$

and update the Gaussian center to $\mu_i = p_i + \Delta\mu_i$.

Inspired by prior work [34, 36] demonstrating the benefits for denser Gaussian placement in regions with richer textures, this drift mechanism ensures each primitive to align with structural boundaries identified in the contour feature space, while retaining consistency with the original semantic features.

3.5 Train GauSAM

We categorize the model parameters into two groups with distinct optimization strategies. The first group includes the pre-trained SAM image encoder and prompt encoder, whose parameters remain frozen to preserve their pretrained capacity. To adapt these pretrained components to our modified model, we introduce trainable LoRA adapters into the query Q and value V projection layers of each attention block, following prior work on efficient model adaptation of SAM [3]. The second group comprises all newly introduced modules, including the contour-guided feature extraction layers and the Gaussian-field decoder. These components are trained from scratch and fully optimized during training. The model is supervised using a pixel-wise *Dice-Cross-Entropy* (*DiceCE*) loss:

$$L_{\text{seg}}(o, \hat{o}) = \alpha L_{\text{CE}}(o, \hat{o}) + \beta L_{\text{Dice}}(o, \hat{o}), \quad (12)$$

where L_{CE} measures the average cross-entropy across all N pixels in the image, L_{Dice} quantifies the spatial overlap between the predicted mask $\hat{o}$ and the ground-truth mask o , and α and β balance their contributions.

4 Experiments

4.1 Datasets

We evaluate GauSAM on two representative benchmarks: Kvasir-Sessile [40] (196 sessile-polyp RGB images) for binary polyp segmentation and BCV [41] (30 CT slices with 13 organ classes) for multi-organ abdominal segmentation. To assess cross-domain generalization, models trained on Kvasir-Sessile are tested on CVC-ClinicDB [42] (612 RGB images from 31 endoscopic sequences), while those trained on BCV are evaluated on AMOS [43] (240 CT cases with liver segmentation only). All 3D CT volumes are processed as 2D axial slices. Data are split into training, validation, and testing sets with a 60:20:20 ratio. Input images are resized to 384×384 for polyp segmentation and 512×512 for multi-organ segmentation. To ensure fair comparison, we follow the preprocessing and evaluation protocols of I-MedSAM [25], reporting Dice Coefficient (Dice) and Hausdorff Distance (HD) on the test sets.

4.2 Implementation Details

Gaussian Field Initialization To enable the continuous Gaussian field formulation, we initialize the Gaussian Parameter Library with 500 learnable templates. Each Gaussian primitive is parameterized by three elements of the covariance matrix Σ , including σ_x, σ_y (linearly spaced between 0.2 and 3.0), and zero correlation, with an initial value of $\sigma(1) \approx 0.73$ for opacity ξ . Primitive centers are placed on a uniform grid, with position-drift offsets initialized to zero. Both the template parameters and the weights of the drift MLP employ Kaiming uniform initialization to ensure stable training.

Model Architecture and Training We adopt the SAM-ViT-B backbone with LoRA adapters (rank 4) inserted into the query and value projections of each transformer block. The “up” projections use Kaiming uniform initialization, while the “down” projections are zero-initialized, ensuring early updates are confined to the LoRA modules, leaving the original SAM weights frozen. Only the LoRA-adapted layers and newly introduced modules are updated. A dual-branch learning rate strategy is used, with LoRA parameters trained at $0.05 \times$ the rate of newly initialized ones. Optimization is performed using AdamW with a weight decay of 10^{-4} and a linear learning rate scheduler. The segmentation head is a lightweight MLP with two hidden layers of dimension 256. The Gaussian Parameters Library contains 500 learnable templates. Training is conducted on 4 NVIDIA H20 GPUs. The polyp segmentation task (single class) requires approximately 3 hours, while the multi-organ segmentation task takes around 50 hours. All models are trained for up to 1000 epochs with early stopping (patience = 10), using a hybrid DiceCE loss. Reported Dice and Hausdorff scores are evaluated on the test set using the checkpoint with the highest validation Dice.

Table 1: Segmentation results on binary polyp and multi-class organ tasks, compared with state-of-the-art discrete and Continuous approaches. Dice (%) denotes the mean Dice score $\pm$ standard deviation over 6 runs (fixed data splits, varying seeds). Params (M) indicates the number of trainable parameters, excluding frozen parameters that are not updated during optimization. Bold highlights the best performance in each task.

Method	Binary Polyp Segmentation		Multi-class Organ Segmentation	
	Dice (%) $\uparrow$	Params (M) $\downarrow$	Dice (%) $\uparrow$	Params (M) $\downarrow$
<i>Discrete Approaches</i>				
U-Net [2]	63.89 $\pm$ 1.30	7.9	74.47 $\pm$ 1.57	16.3
Res2UNet [44]	81.62 $\pm$ 0.97	25.4	79.23 $\pm$ 0.66	38.3
nnUNet [16]	82.97 $\pm$ 0.89	126.6	85.15 $\pm$ 0.67	126.6
MedSAM [45]	82.88 $\pm$ 0.55	4.1	85.85 $\pm$ 0.81	52.7
<i>Continuous Approaches</i>				
OSSNet [46]	76.11 $\pm$ 1.14	5.2	73.38 $\pm$ 1.65	7.6
IOSNet [7]	78.37 $\pm$ 0.76	4.1	76.75 $\pm$ 1.37	6.2
SwIPE [47]	85.05 $\pm$ 0.82	2.7	81.21 $\pm$ 0.94	4.4
I-MedSAM [25]	91.49 $\pm$ 0.52	1.6	89.91$\pm$0.68	3.5
GauSAM(Ours)	94.76$\pm$0.32	1.2	89.75 $\pm$ 0.47	2.9

4.3 Quantitative Results

Comparison with State-of-the-Art Methods We evaluate GauSAM on two representative benchmarks: the Kvasir-Sessile dataset for binary polyp segmentation, and the BCV dataset for multi-class abdominal organ segmentation. The compared methods are grouped into two categories: discrete approaches, which predict segmentation masks on fixed grids, and continuous approaches, which represent masks in resolution-agnostic or implicit forms. Discrete baselines include U-Net [2], Res2UNet [44], nnUNet [16], and MedSAM [45]. Continuous baselines include OSSNet [46], IOSNet [7], SwIPE [47], and I-MedSAM [25] — the latter replaces the original decoder of SAM with an INR-based formulation, and represents the previous state-of-the-art.

As demonstrated in Table 1, GauSAM consistently achieves the best performance across both tasks, obtaining the highest Dice scores while maintaining the lowest number of trainable parameters. On the Kvasir-Sessile dataset, GauSAM reaches a Dice of **94.76%**, outperforming I-MedSAM [25] by over 3 points with fewer parameters (1.2M vs. 1.6M). On the multi-organ task, GauSAM achieves a Dice of **89.75%**, which is on par with I-MedSAM, despite using only 2.9M parameters.

Cross-Domain Generalization Performance We evaluate the cross-domain generalizability of GauSAM under two settings: binary-class polyp segmentation from Kvasir-Sessile to CVC, and from multi-organ segmentation in BCV to liver-only segmentation in AMOS. All models are trained on the source domain and directly evaluated on the target domain without any fine-tuning. As shown in Table 2, GauSAM achieves the highest Dice scores across both transfer scenarios, reaching **91.79%** on CVC and **90.62%** on AMOS. Compared with discrete and continuous baselines, GauSAM exhibits superior generalization under domain shifts. In particular, it consistently outperforms the previous state-of-the-art I-MedSAM [25] and other baselines.

Cross-Resolution Generalization Performance To evaluate the resolution-agnostic capability of GauSAM, we conduct cross-resolution segmentation experiments on the Kvasir-Sessile dataset. All models are trained with input images and masks at 384×384 , and are then evaluated by producing output masks at 128×128 and 896×896 , without any fine-tuning or architectural modification. As shown in Table 3, GauSAM consistently achieves best performance under both downsampling and upsampling conditions, reaching Dice scores of **94.35%** and **94.29%**, respectively. In contrast, discrete baselines exhibit notable performance drops when deployed at unseen resolutions, and even implicit models such as SwIPE [47] and I-MedSAM [25] show limited adaptability.

Boundary Comparison Hausdorff Distance (HD) measures boundary accuracy by capturing the largest surface deviation between the prediction and the ground truth. As shown in Table 4, GauSAM achieves strong performance across most settings, particularly in cross-resolution scenarios. A relatively weaker HD score is observed in the domain adaptation from Kvasir to CVC; however, a significant improvement is still evident in the corresponding Dice score as shown in Table 2. In the

Table 2: Cross-domain segmentation results from Kvasir-Sessile to CVC and from BCV to AMOS.

Method	Dice (%) $\uparrow$
<i>Kvasir-Sessile $\rightarrow$ CVC</i>	
nnUNet [16]	84.91
MedSAM [45]	74.59
I-MedSAM [25]	88.83
GauSAM (Ours)	91.79
<i>BCV $\rightarrow$ AMOS</i>	
nnUNet [16]	79.63
MedSAM [45]	71.98
I-MedSAM [25]	86.28
GauSAM (Ours)	90.62

Table 3: Cross-resolution segmentation results on Kvasir-Sessile from 384 to 128 and 384 to 896. (*: adjusted input resolution)

Method	Dice (%) $\uparrow$	
	384 $\rightarrow$ 128	384 $\rightarrow$ 896
<i>Discrete Approaches</i>		
nnUNet [16]	73.97	83.56
nnUNet* [16]	65.34	76.36
MedSAM [45]	82.39	83.19
MedSAM* [45]	82.37	83.32
<i>Implicit Approaches</i>		
SwIPE [47]	81.26	84.33
I-MedSAM [25]	91.45	91.33
GauSAM (Ours)	94.35	94.29

Table 4: Hausdorff Distance comparison on various experiment settings.

HD distance ($\downarrow$)	Kvasir-Sessile	Kvasir-Sessile $\rightarrow$ CVC	384 $\rightarrow$ 128	384 $\rightarrow$ 896	BCV	BCV $\rightarrow$ AMOS
nnUNet [16]	31.30	82.31	13.69	72.31	6.50	80.39
MedSAM [45]	21.53	30.15	8.04	51.82	10.62	52.14
I-MedSAM [25]	11.59	19.76	7.91	32.77	5.95	37.53
GauSAM (Ours)	10.29	20.43	4.21	22.94	11.21	36.51

Table 5: End-to-end inference performance comparison. Parameters (Params) denote the total number of model parameters in millions (M). GFLOPs represent giga floating-point operations. Latency is the average inference time in milliseconds (ms) with standard deviation (STD). FPS indicates frames per second. Memory refers to peak GPU memory usage in megabytes (MB). All metrics were measured on a single NVIDIA RTX 4090 GPU with batch size 1 and input resolution 512×512 , averaged over 100 forward passes after 10 warm-ups.

Model	Params (M)	GFLOPs	Latency (ms) $\pm$ STD	FPS	Memory (MB)
U-Net	31.03	218.97	41 $\pm$ 0.06	24.4	1244
MedSAM	93.74	573.87	66 $\pm$ 0.91	15.2	2768
I-MedSAM	92.88	648.06	86 $\pm$ 3.12	11.6	3021
GauSAM (Ours)	92.51	473.35	188 $\pm$ 2.34	5.3	1898
GauSAM w/o NSCT	92.51	374.91	79 $\pm$ 2.15	12.7	1898

BCV source-domain setting, GauSAM does not perform well, which may be attributed to the use of a single continuous Gaussian field for all organ classes, potentially limiting precision on small structures in multi-organ segmentation.

Computational Efficiency To evaluate the computational efficiency of GauSAM, we benchmark it against several representative baselines. As demonstrated in Table 5, GauSAM exhibits lower computational efficiency, while delivering superior segmentation performance. We emphasize that this inference speed is well within the practical range for medical applications, fully meeting the real-time demands of clinical workflows. In medical image segmentation tasks, precision is always paramount, as accurate delineation of anatomical structures or pathological regions directly impacts diagnostic accuracy and patient safety. Moreover, empirical evidence shows that removing the NSCT module leads to a substantial drop in GFLOPs and latency, confirming that this component is the main contributor to GauSAM’s computational overhead. The Appendix D provides component-wise efficiency metrics and a diagnostic analysis of the NSCT module’s comparatively lower efficiency.

4.4 Qualitative Results

Figure 3 presents representative qualitative comparisons on both binary and multi-class segmentation tasks. We visualize the predicted segmentation masks of different models by overlaying them on the

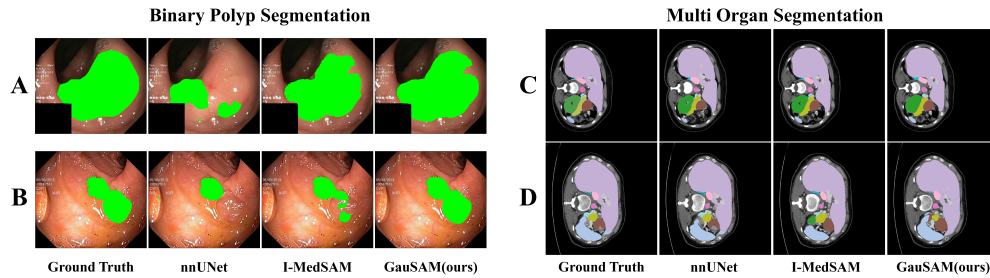


Figure 3: Qualitative comparison on binary polyp segmentation and multi-organ segmentation.

original input images. Compared with existing discrete and implicit methods, GauSAM produces masks that exhibit significantly improved semantic smoothness and structural consistency. These improvements are particularly evident in *Sample A*, where GauSAM demonstrates superior edge continuity along the polyp's right boundary compared to the previous SOTA method, I-MedSAM, which produces fragmented artifacts. In contrast, GauSAM maintains consistent boundaries without such discontinuities. These results are largely attributed to our continuous Gaussian field formulation, which leverages elliptical Gaussian kernel to ensure spatial continuity and local consistency, meaning that adjacent pixels are more likely to belong to the same object. This principled design, combined with contour-guided structural cues, enables the model to better capture fine-grained geometry and align predictions with anatomical boundaries, effectively mitigating discretization artifacts and enhancing the coherence of structural delineation across diverse regions.

4.5 Ablation Study

Our core contribution lies in the design of the continuous Gaussian Field (GS), upon which all subsequent modules are built. Therefore, we conduct ablation studies on the Kvasir-Sessile dataset under three settings: in-domain, cross-domain, and cross-resolution. As shown in Table 6, the baseline model without GS performs significantly worse across all tasks. After introducing the Gaussian Field alone, the model yields substantial improvements, validating its critical role. Further integrating the Contour Propagation Unit (CPU) enhances spatial consistency, while the Contour-Guided Position Drift (PosOff) improves robustness under resolution shifts. The full model, which includes all three components, consistently achieves the highest Dice scores in every setting.

Table 6: Ablation study on the effectiveness of each component in the pipeline. Dice(%) is reported on in-domain, cross-domain, and cross-resolution settings.

GS	CPU	PosOff	Kvasir-Sessile	Cross-domain Kvasir $\rightarrow$ CVC	Cross-resolution	
					384 $\rightarrow$ 128	384 $\rightarrow$ 896
–	–	–	86.77	84.51	75.02	77.43
✓	–	–	92.62	91.00	91.44	92.74
✓	✓	–	93.62	91.07	92.33	94.00
✓	✓	✓	94.76	91.79	94.35	94.29

5 Conclusion and Discussion

In this paper, we presents GauSAM, a novel segmentation framework that combines contour-enhanced encoding with continuous Gaussian field decoding. By constructing a continuous resolution-agnostic Gaussian probability field, our method addresses limitations of structural precision and boundary localization of both discrete and INR-based approaches. Extensive experiments across binary and multi-class segmentation, under in-domain, cross-domain, and cross-resolution settings, demonstrate GauSAM's superior performance in terms of semantic consistency and edge accuracy. However, GauSAM underperforms on multi-class segmentation benchmarks, likely due to its use of a single continuous field to represent multiple classes, which may obscure fine-grained details of small objects. Despite this, GauSAM is the first to model SAM's segmentation probability field as a continuous representation parameterized by 2D Gaussian Splatting for mask generation and establishes a strong foundation for future improvements in multi-class scenarios.

References

- [1] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32:582–596, 2019.
- [2] 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, pages 234–241. Springer, 2015.
- [3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.
- [4] Fang Chen, Lingyu Chen, Haojie Han, Sainan Zhang, Daoqiang Zhang, and Hongen Liao. The ability of segmenting anything model (sam) to segment ultrasound images. *BioScience Trends*, 17(3):211–218, 2023.
- [5] Yusuke Sugawara, Sayaka Shiota, and Hitoshi Kiya. Checkerboard artifacts free convolutional neural networks. *APSIPA Transactions on Signal and Information Processing*, 8:e9, 2019.
- [6] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.
- [7] Muhammad Osama Khan and Yi Fang. Implicit neural representations for medical imaging segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 433–443. Springer, 2022.
- [8] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.
- [9] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, 2018. URL <https://arxiv.org/abs/1802.10062>.
- [10] Rohan Asthana, Joschua Conrad, Maurits Ortmanns, and Vasileios Belagiannis. Dextr: Zero-shot neural architecture search with singular value decomposition and extrinsic curvature, 2025. URL <https://arxiv.org/abs/2508.12977>.
- [11] Alban Desmaison, Rudy Bunel, Pushmeet Kohli, Philip H. S. Torr, and M. Pawan Kumar. Efficient continuous relaxations for dense crf, 2016. URL <https://arxiv.org/abs/1608.06192>.
- [12] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models, 2022. URL <https://arxiv.org/abs/2210.02025>.
- [13] Yan Zhou, Bo Dong, Yuanfeng Wu, Wentao Zhu, Geng Chen, and Yanning Zhang. Dichotomous image segmentation with frequency priors. In *IJCAI*, pages 1822–1830, 2023.
- [14] Arthur L Da Cunha, Jianping Zhou, and Minh N Do. The nonsubsampling contourlet transform: theory, design, and applications. *IEEE transactions on image processing*, 15(10):3089–3101, 2006.
- [15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [16] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

- [17] 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.
- [18] Jieneng Chen, Jieru Mei, Xianhang Li, Yongyi Lu, Qihang Yu, Qingyue Wei, Xiangde Luo, Yutong Xie, Ehsan Adeli, Yan Wang, et al. Transunet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers. *Medical Image Analysis*, page 103280, 2024.
- [19] 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.
- [20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [21] Tristan Piater, Björn Barz, and Alexander Freytag. Prompt-tuning sam: From generalist to specialist with only 2048 parameters and 16 training images. *arXiv preprint arXiv:2504.16739*, 2025.
- [22] Mingzhou Jiang, Jiaying Zhou, Junde Wu, Tianyang Wang, Yueming Jin, and Min Xu. Uncertainty-aware adapter: Adapting segment anything model (sam) for ambiguous medical image segmentation. *arXiv preprint arXiv:2403.10931*, 2024.
- [23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [24] Christoph Reich, Tim Prangemeier, Özdemir Cetin, and Heinz Koepl. OSS-Net: Memory Efficient High Resolution Semantic Segmentation of 3D Medical Data. In *British Machine Vision Conference*. British Machine Vision Association, 2021.
- [25] Xiaobao Wei, Jiajun Cao, Yizhu Jin, Ming Lu, Guangyu Wang, and Shanghang Zhang. I-medsam: Implicit medical image segmentation with segment anything. In *European Conference on Computer Vision*, pages 90–107. Springer, 2024.
- [26] Amirali Molaei, Amirhossein Aminimehr, Armin Tavakoli, Amirhossein Kazerouni, Bobby Azad, Reza Azad, and Dorit Merhof. Implicit neural representation in medical imaging: A comparative survey. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2381–2391, 2023.
- [27] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106, 2005.
- [28] Fei Zhou, ZhenHong Jia, Jie Yang, and Nikola Kasabov. Method of improved fuzzy contrast combined adaptive threshold in nsct for medical image enhancement. *BioMed research international*, 2017(1):3969152, 2017.
- [29] Ehsan Nezhadarya and Mohammad B Shamsollahi. Image contrast enhancement by contourlet transform. In *Proceedings ELMAR 2006*, pages 81–84. IEEE, 2006.
- [30] Reza Darooei, Milad Nazari, Rahele Kafieh, and Hossein Rabbani. Optimal deep learning architecture for automated segmentation of cysts in oct images using x-let transforms. *Diagnostics*, 13(12):1994, 2023.
- [31] Imene Mecheter, Maysam Abbod, Abbes Amira, and Habib Zaidi. Deep learning with multiresolution handcrafted features for brain mri segmentation. *Artificial intelligence in medicine*, 131: 102365, 2022.
- [32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

- [33] Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. Gaussiansr: High fidelity 2d gaussian splatting for arbitrary-scale image super-resolution, 2024. URL <https://arxiv.org/abs/2407.18046>.
- [34] Long Peng, Anran Wu, Wenbo Li, Peizhe Xia, Xueyuan Dai, Xinjie Zhang, Xin Di, Haoze Sun, Renjing Pei, Yang Wang, et al. Pixel to gaussian: Ultra-fast continuous super-resolution with 2d gaussian modeling. *arXiv preprint arXiv:2503.06617*, 2025.
- [35] Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and efficient 2d gaussian splatting for arbitrary-scale super-resolution. *arXiv preprint arXiv:2501.06838*, 2025.
- [36] Mufan Liu, Qi Yang, Miaoran Zhao, He Huang, Le Yang, Zhu Li, and Yiling Xu. D2gv: Deformable 2d gaussian splatting for video representation in 400fps. *arXiv preprint arXiv:2503.05600*, 2025.
- [37] Mihnea-Bogdan Jurca, Remco Royen, Ion Giosan, and Adrian Munteanu. Rt-gs2: Real-time generalizable semantic segmentation for 3d gaussian representations of radiance fields. *arXiv preprint arXiv:2405.18033*, 2024.
- [38] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.
- [39] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024.
- [40] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *International conference on multimedia modeling*, pages 451–462. Springer, 2019.
- [41] Bennett Landman, Zhoubing Xu, Juan Igelsias, Martin Styner, Thomas Langerak, and Arno Klein. Miccai multi-atlas labeling beyond the cranial vault—workshop and challenge. In *Proc. MICCAI multi-atlas labeling beyond cranial vault—workshop challenge*, volume 5, page 12. Munich, Germany, 2015.
- [42] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized medical imaging and graphics*, 43: 99–111, 2015.
- [43] Yuanfeng Ji, Haotian Bai, Chongjian Ge, Jie Yang, Ye Zhu, Ruimao Zhang, Zhen Li, Lingyan Zhanng, Wanling Ma, Xiang Wan, et al. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. *Advances in neural information processing systems*, 35:36722–36732, 2022.
- [44] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.
- [45] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15:654, 2024.
- [46] Christoph Reich, Tim Prangemeier, Özdemir Cetin, and Heinz Koepl. Oss-net: memory efficient high resolution semantic segmentation of 3d medical data. *arXiv preprint arXiv:2110.10640*, 2021.
- [47] Yejia Zhang, Pengfei Gu, Nishchal Sapkota, and Danny Z Chen. Swipe: Efficient and robust medical image segmentation with implicit patch embeddings. In *International conference on medical image computing and computer-assisted intervention*, pages 315–326. Springer, 2023.

A Additional Analysis of NSCT and Related Modules

The integration of the Non-Subsampled Contourlet Transform (NSCT) into the GauSAM framework plays a pivotal role in enhancing the segmentation accuracy of the Segment Anything Model (SAM) for both multi-organ and binary polyp segmentation tasks. The choice of NSCT and the design of related modules stems from a meticulous combination of theoretical analysis and extensive experimental validation, as detailed in the following subsections.

A.1 Formal Analysis of Contour-Guided Position Drift

The Contour-Guided Position Drift module improves boundary alignment by applying bounded, differentiable shifts to the centers of Gaussian kernels. For the i -th anisotropic Gaussian kernel:

$$G_i(p \mid \mu_i, \Sigma_i) = \frac{1}{2\pi|\Sigma_i|} \exp\left(-\frac{1}{2}(p - \mu_i)^\top \Sigma_i^{-1}(p - \mu_i)\right), \quad (13)$$

its center is updated from the initial position p_i by:

$$\mu_i = p_i + \Delta\mu_i, \quad \Delta\mu_i = \tanh(\text{MLP}_{\text{drift}}(u_i)), \quad (14)$$

where $\tanh(\cdot)$ ensures $\|\Delta\mu_i\| < 1$ for stability. During training, according to the chain rule, the gradient of the segmentation loss L with respect to μ_i is:

$$\frac{\partial L}{\partial \mu_i} = \int_{\Omega} \frac{\partial L}{\partial X(p)} \frac{\partial X(p)}{\partial G_i} \frac{\partial G_i}{\partial \mu_i} dp, \quad (15)$$

with

$$\frac{\partial G_i(p)}{\partial \mu_i} = \Sigma_i^{-1}(p - \mu_i) G_i(p). \quad (16)$$

This formulation indicates that a small adjustment of μ_i redistributes the kernel's density in a distance-weighted manner, directly influencing how features are aligned to local structures. Through the fully differentiable mapping $\Delta\mu_i = \tanh(\text{MLP}_{\text{drift}}(u_i))$, these gradients propagate to the parameters of the drift network, enabling the drift module to automatically optimize kernel placement. Consequently, the drift module is a mathematically grounded, bounded, and differentiable mechanism that, under gradient descent, provably minimizes boundary misalignment, complementing NSCT's precise contour extraction.

A.2 Qualitative Analysis of NSCT-Processed Edge Maps

To further elucidate the contribution of NSCT, we present an additional qualitative analysis based on the edge maps generated from NSCT processing, as illustrated in Figure 4. The provided image showcases a comparative visualization of original images and their corresponding NSCT-processed edge maps across two distinct medical imaging modalities: CT scans for multi organ segmentation (rows 1-2) and endoscopic images for binary polyp segmentation (rows 3-4). In the original CT images (row 1), organ boundaries exhibit significant ambiguity due to low contrast and overlapping structures, posing challenges for precise delineation. Similarly, the original endoscopic images (row 3) reveal blurred polyp edges, often confounded by background noise and varying illumination. The application of NSCT, however, effectively transforms these inputs into well-defined edge maps (rows 2 and 4), where anatomical boundaries—such as those between organs or around polyps—are sharply delineated with enhanced multidirectional frequency information. This enhancement lays a solid foundation for GauSAM's encoding process in terms of edge information representation and precise Gaussian kernel distribution, a benefit also reflected in the ablation study results, particularly in cross-domain and cross-resolution tasks.

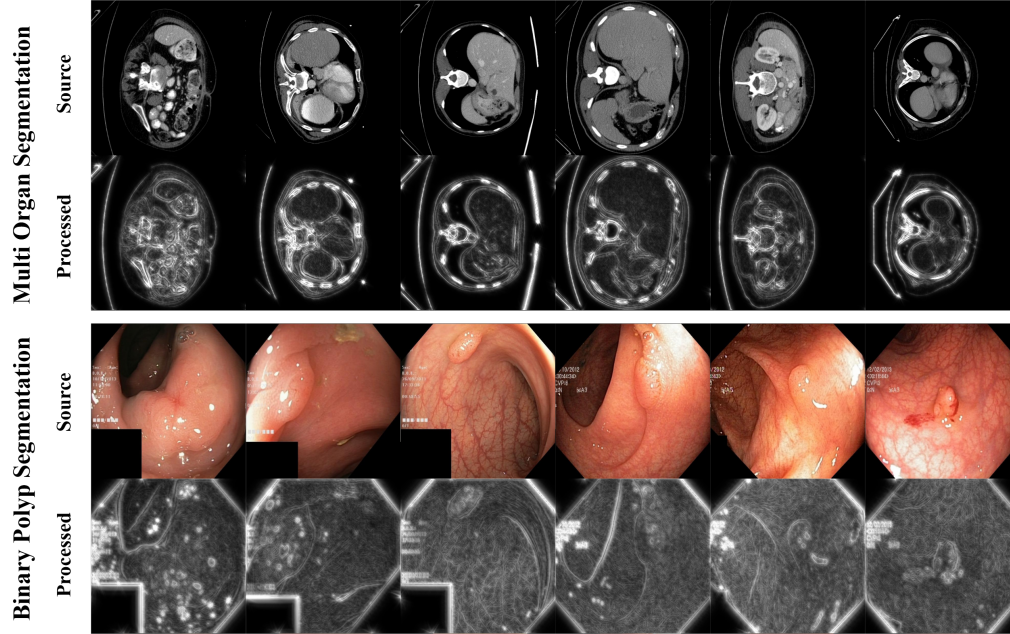


Figure 4: Qualitative comparison of source images (top) and their corresponding NSCT-processed edge maps (bottom) for multi-organ segmentation (rows 1-2) and binary polyp segmentation (rows 3-4). The NSCT-processed edge maps enhance boundary clarity, improving SAM’s encoder edge representation and guiding the decoder’s Gaussian kernel distribution.

Table 7: Symbol definitions for Gaussian Parameters Library and traditional Gaussian Splatting.

Symbol	Definition
i	Index of each Gaussian kernel
k	Index of each template in the library
N	Number of Gaussian kernels
m	Number of templates in the library ($N \gg m$)
Σ_i	2×2 symmetric covariance matrix for kernel i , controlling ellipsoid shape and orientation (3 parameters)
ξ_i	Opacity of kernel i (1 parameter)
$\Sigma^{(k)}$	Covariance matrix of template k
$\xi^{(k)}$	Opacity of template k
θ_i	Parameters of kernel i , including Σ_i and ξ_i
θ	All parameters of traditional method, $\{\theta_i\}_{i=1}^N$
ϕ	Library parameters, $\{(\sigma_x/\sigma_y, \text{corr}, \xi^{(k)})\}_{k=1}^m$

B Detailed Analysis of Gaussian Parameters Library

To provide a comprehensive understanding of the Gaussian Parameters Library, we present a formal comparison with the traditional Gaussian Splatting method, highlighting the theoretical and practical advantages of our approach. The adoption of the Gaussian Parameters Library is grounded in an extensive literature review, rigorous theoretical derivations, and comprehensive experimental validation, as evidenced by the ablation study on library size reported in Section E.2. The library approach achieves superior training efficiency and stability compared to traditional Gaussian Splatting, which we formalize below. Due to fundamental architectural differences, direct ablation of the library is impractical, as it would necessitate a complete restructuring of the model. The following subsections define the key symbols and provide a detailed comparison of the two approaches.

B.1 Symbol Definitions

Table 7 defines the symbols used in the analysis of the Gaussian Parameters Library and traditional Gaussian Splatting.

B.2 Comparison with Traditional Gaussian Splatting

The Gaussian Parameters Library optimizes the covariance matrix Σ_i and opacity ξ_i of Gaussian kernels with greater stability and efficiency than traditional Gaussian Splatting. Below, we compare the two approaches across several key aspects, focusing on parameter count, parameter space, initialization and training stability, learning objective, and model structure.

Parameter Count Classical Gaussian Splatting constructs a continuous field by overlapping many elliptical kernels so that, after opacity accumulation, the synthesized values at discrete pixels/feature locations match the observed signal. Training therefore adjusts the kernels' shapes and opacities to a specific image so as to best reconstruct its discrete measurements, i.e., an instance-specific optimization without parameter sharing across images. In practice, the number of kernels N scales with the number of feature-space locations (often one kernel per pixel/voxel/patch), so the parameter size grows with image resolution/content. Without any library/shared parameterization, this instance-specific paradigm necessitates retraining for each new image, which is computationally prohibitive at scale.

Formally, if each of the N kernels uses a 2×2 symmetric covariance Σ_i (3 parameters) and an opacity ξ_i (1 parameter), the parameter count is

$$P_{\text{trad}} = (3 + 1) \cdot N = 4N. \quad (17)$$

By contrast, we introduce a template library with m shared kernels and a lightweight selection head, which inherently form a hypernetwork. Given image embeddings, the hypernetwork maps image features directly to each kernel's parameters by selecting (and modulating) a template for that location, thus enabling runtime parameter prediction and avoiding instance-specific retraining. The library holds only m sets of covariance/opacity, yielding

$$P_{\text{lib}} = (3 + 1) \cdot m = 4m. \quad (18)$$

Since typically $N \gg m$, we have $P_{\text{trad}} \gg P_{\text{lib}}$, achieving a substantial reduction in parameter size while enabling end-to-end learning with shared parameters across images.

Parameter Space The traditional method optimizes parameters in a high-dimensional space

$$S_{\text{trad}} = \mathbb{R}^{4N}, \quad (19)$$

whereas the library method operates in

$$S_{\text{lib}} = \mathbb{R}^{4m}. \quad (20)$$

Given $N \gg m$, the lower-dimensional space of the library method facilitates faster convergence and reduces computational complexity.

Initialization and Training Stability The traditional method employs random initialization for $\theta_i = (\Sigma_i, \xi_i)$, which risks degenerate covariance matrices (i.e., $\det(\Sigma_i) \approx 0$), causing ellipse collapse and unstable gradients. The library method initializes

$$\phi = \{(\sigma_x/\sigma_y) \in [0.2, 3.0], \text{corr} = 0, \xi^{(k)} = \text{sigmoid}(1)\} \quad (21)$$

with Kaiming uniform initialization and uses Gumbel-Softmax to select from m positive definite templates $\{\Sigma^{(k)}\}_{k=1}^m$. This ensures valid Σ_i values, enhancing initialization stability and training robustness.

Learning Objective and Training Efficiency The traditional method optimizes the loss

$$L_{\text{trad}} = \min_{\theta \in \mathbb{R}^{4N}} L(\theta), \quad (22)$$

which is computationally intensive due to the high-dimensional space. The library method optimizes

$$L_{\text{lib}} = \min_{\phi \in \mathbb{R}^{4m}} L(\phi), \quad (23)$$

where a selection head with Gumbel-Softmax assigns one of the m templates to each kernel, operating in the lower-dimensional space $\mathbb{R}^{4m}$. This constrained optimization accelerates convergence and improves training efficiency.

C Evaluation Results on Additional Datasets

Due to strict page constraints in the main manuscript, we presented quantitative results only on the Kvasir-Sessile and BCV datasets, selected for their representativeness and recognized challenges in medical image segmentation. To further demonstrate the robustness of GauSAM, we conducted extensive experiments across additional medical imaging modalities and segmentation tasks. In this section, we report supplementary results on the REFUGE2 dataset (color fundus images for optic cup and disc segmentation) and the Breast Ultrasound Image (BUSI) dataset (lesion segmentation), which further validate GauSAM’s state-of-the-art performance across diverse tasks and modalities. As shown in Table 8, GauSAM consistently outperforms baseline methods, achieving the highest Dice scores across all tasks. On the REFUGE2 dataset, GauSAM attains Dice scores of 89.1% for optic cup segmentation and 96.2% for optic disc segmentation, surpassing the previous state-of-the-art method, I-MedSAM, by 1.2% and 2.2%, respectively. Similarly, on the BUSI dataset, GauSAM achieves a Dice score of 92.16% for lesion segmentation, outperforming I-MedSAM by 0.61%. These results confirm GauSAM’s superior robustness in handling diverse medical imaging challenges.

Table 8: Performance comparison on REFUGE2 and BUSI datasets. Dice (%) is reported for each task, evaluated on the respective test sets.

Model	Optic Cup Dice (%)	Optic Disc Dice (%)	BUSI Lesion Dice (%)
U-Net [2]	82.3	92.1	67.24
nnUNet [16]	84.9	94.7	75.19
I-MedSAM [25]	87.9	94.0	91.55
GauSAM (Ours)	89.1	96.2	92.16

D Detailed Computational Efficiency Analysis of GauSAM Components

To further elucidate the computational efficiency of GauSAM, we provide a detailed decomposition of its component-level inference performance in Table 9. The table presents the computational metrics for each major component of GauSAM, and the evaluation protocol is identical to that of Section 4.3. As shown in Table 9, the NSCT module accounts for approximately 58% of GauSAM’s total latency (109 ms out of 188 ms), making it the primary contributor to the model’s computational overhead.

This phenomenon can be attributed to two key factors:

1. **NSCT Feature Richness vs. Computational Cost:** The NSCT module excels in capturing multi-scale and multi-directional contour features, which significantly enhance segmentation accuracy (e.g., a 1.08% Dice improvement over FFT, 94.76% vs. 93.68%). However, its shift-invariant filter bank decomposition across multiple scales and orientations inherently increases computational complexity, leading to higher GFLOPs and latency.
2. **Implementation Limitations:** Existing NSCT implementations are outdated and limited to Python 2.x, lacking compatibility with modern architectures. Our custom Python 3.x implementation, while supporting vectorized processing for batch sizes greater than 1, lacks

the low-level optimizations (e.g., C++ or CUDA-based libraries) available in more mature algorithms, resulting in suboptimal execution efficiency.

These findings confirm that the computational overhead in GauSAM is primarily due to the NSCT module’s intrinsic complexity and suboptimal implementation. Nevertheless, the enhanced contour accuracy justifies this trade-off, as it ensures robust representational quality critical for medical imaging while maintaining clinically acceptable throughput.

Table 9: Component-wise inference profile of GauSAM, under the same metric definitions and evaluation protocol as Table 5.

Component	Params (M)	GFLOPs	Latency (ms) $\pm$ STD	FPS	Memory (MB)
NSCT	≈ 0	98.44	109.01 ± 1.82	9.2	113
Contour-Enhanced Encoder	92.13	220.18	43.53 ± 0.79	23.0	388
2D Gaussian Continuous Field	0.38	154.73	32.70 ± 1.15	30.6	1397
GauSAM (end-to-end)	92.51	473.35	188.01 ± 2.34	5.3	1898

E Additional Ablation Studies

E.1 Contour Extraction Ablation

To validate GauSAM’s choice of the Non-Subsampled Contourlet Transform (NSCT) for contour feature extraction, we conduct ablation studies on the Kvasir-Sessile dataset, replacing NSCT with alternative contour/high-frequency methods: FFT (as used in I-MedSAM [25]), Canny, Wavelet, and Sobel. As shown in Table 10, the full GauSAM with NSCT achieves the highest Dice score and lowest Hausdorff Distance (HD), confirming its superiority in precise boundary delineation. Unlike FFT’s broad high-frequency components that include noise, NSCT selectively captures semantically meaningful contours, aligning directly with the requirements of semantic segmentation and providing superior boundary representation. These experiments confirm NSCT’s superiority in precise boundary delineation and its better compatibility with GauSAM.

Table 10: Ablation study on contour feature extraction methods on Kvasir-Sessile.

Method	Dice (%)	HD
FFT	93.68	11.34
Canny	93.66	11.82
Wavelet	94.13	11.40
Sobel	93.71	11.15
NSCT (Ours)	94.76	10.29

E.2 Impact of Gaussian Parameter Library Size

As reported in Table 11, expanding the Gaussian Parameter library from $m = 50$ to $m = 500$ leads to a marked improvement in segmentation accuracy and boundary precision. This improvement indicates that a richer set of 2D Gaussian templates better captures the diverse scale and shape variations in our medical image benchmarks. Beyond $m = 500$, however, the benefit plateaus and slightly reverses, indicating diminishing returns once the library exceeds a critical size.

A primary factor behind this saturation is the linear growth of the selection head’s parameters with m . Since the module must assign a probability to each template via a soft selection mechanism, overly large m introduces many logits that receive relatively weak gradient signals, leading to under-utilized or poorly distinguished templates and impeding stable convergence during training.

In addition, an excessively large library incurs higher computational and memory costs at both training and inference time, while introducing substantial redundancy: many Gaussian templates end up modeling similar feature patterns, so their incremental contribution becomes marginal. Based on a series of controlled experiments, we find $m = 500$ strikes the best balance between expressive power and learnability, yielding the highest average Dice score with moderate overhead.

Table 11: Segmentation performance under different Gaussian parameter library sizes on the Kvasir-Sessile dataset.

Library size m	50	100	300	500 (default)	800	1000
Dice (%)	92.93	93.78	94.21	94.76	94.22	94.43
HD Distance	12.53	11.65	11.01	10.29	11.19	10.93
Params (M)	0.96	0.99	1.11	1.22	1.40	1.51

F Supplementary Analysis of Robustness on Low-Quality Images

To further validate its robustness, particularly on low-quality images with blurred boundaries, we conducted additional experiments on the Kvasir-Sessile dataset, comparing GauSAM against the previous state-of-the-art model, I-MedSAM [25]. This analysis provides comprehensive empirical evidence of GauSAM’s superior performance across varying image quality levels. We identified 12 test images where I-MedSAM exhibited relatively low performance ($Dice < 89\%$), all of which displayed ambiguous boundaries. Based on this observation, we partitioned the test set into two groups: a *High Ambiguity* subset, comprising images with blurred or uncertain boundaries, and a *Normal* subset, consisting of images with clear, well-defined boundaries. Table 12 presents the mean Dice scores for both models on these subsets.

As shown in Table 12, GauSAM achieves a substantial Dice improvement of +7.3% over I-MedSAM on the High Ambiguity subset, compared to a +1.9% improvement on the Normal subset. This pronounced performance gain in the ambiguous boundary scenario underscores GauSAM’s robustness in handling low-quality images with blurred boundaries. Furthermore, GauSAM exhibits remarkable consistency across image quality levels, with only a 0.5% Dice score difference between the High Ambiguity and Normal subsets, in contrast to I-MedSAM’s 6% gap. These results confirm GauSAM’s ability to maintain stable and accurate segmentation performance across diverse image quality conditions, reinforcing its effectiveness for both challenging and well-defined cases.

Table 12: Dice performance comparison of GauSAM and I-MedSAM on the Kvasir-Sessile test set, divided into High Ambiguity and Normal image groups. Evaluations were conducted with a batch size of 1 and input resolution of 512×512 . Dice Gain represents the improvement of GauSAM over I-MedSAM.

Image Type	Sample Count	I-MedSAM Dice (%)	GauSAM Dice (%)	Dice Gain (%)
High Ambiguity	11	86.8	94.1	+7.3
Normal	28	92.7	94.6	+1.9

G Extension to 3D Segmentation Tasks

The design of GauSAM, particularly its use of 2D Gaussian Splatting and contour-guided mechanisms, is inherently extensible to 3D volumetric segmentation. This section provides a formal analysis of how GauSAM’s framework can be adapted to 3D tasks, highlighting its potential to enable expressive and spatially coherent volumetric representations while preserving fine-grained anatomical details. The following subsections extend the core components of GauSAM—specifically the Non-Subsampled Contourlet Transform (NSCT) and the Gaussian Continuous Probability Field—to their 3D counterparts, demonstrating the theoretical compatibility and practical promise of this extension.

G.1 3D Non-Subsampled Contourlet Transform

In 3D, the input volume is represented as $I \in \mathbb{R}^{B \times C \times D \times H \times W}$, where D denotes the depth dimension. The Non-Subsampled Contourlet Transform (NSCT) is extended to 3D-NSCT, decomposing the volume into multi-scale, multi-directional subbands:

$$\{C_{l,\theta,\phi}\}_{l=1}^{L,\Theta_l,\Phi_l} = \mathcal{NSCT}^{3D}(I), \quad (24)$$

where Θ and Φ represent angular directions in 3D space. These subbands capture volumetric contour features, enhancing boundary awareness in 3D segmentation tasks.

G.2 3D Gaussian Continuous Probability Field

The 2D Gaussian probability field is extended to 3D by defining a 3D Gaussian primitive at each voxel (x, y, z) :

$$g(\mathbf{p}) = \alpha \exp \left(-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{p} - \boldsymbol{\mu}) \right), \quad (25)$$

where $\mathbf{p} = [x, y, z]^T$, $\boldsymbol{\mu} = [\mu_x, \mu_y, \mu_z]^T$ is the mean vector, $\boldsymbol{\Sigma}$ is a 3×3 covariance matrix, and α is the amplitude. The continuous Gaussian field is constructed as:

$$P(\mathbf{p}) = \sum_{i=1}^N w_i g_i(\mathbf{p}), \quad (26)$$

where $N = D \times H \times W$ is the number of voxels, and w_i are learnable weights. The Contour-Guided Position Drift module adjusts the positions of the primitives based on 3D-NSCT subbands, ensuring precise boundary alignment in volumetric data. This formulation demonstrates GauSAM's theoretical compatibility with 3D tasks, leveraging the same principles of spatial continuity and local consistency to achieve robust segmentation performance.

G.3 Adaptability to 3D Segmentation

The successful application of 2D Gaussian Splatting in GauSAM, combined with its contour-guided mechanism, provides a robust foundation for extending the framework to 3D scenarios. The 3D-NSCT enhances boundary fidelity through multi-scale, multi-directional structural priors, while the 3D Gaussian field ensures spatial continuity and local consistency in volumetric data. These adaptations enable GauSAM to model complex 3D medical imaging tasks effectively, facilitating the integration of multi-modal data and enhancing diagnostic precision across varied anatomical contexts.

H Further Discussion

H.1 GauSAM's Strengths in Cross-Domain Generalization and Resolution-Agnostic Inference

GauSAM delivers exceptional performance in both binary polyp segmentation and multi-organ segmentation, excelling in cross-domain and cross-resolution scenarios, which achieves the highest Dice scores and ensures consistency. While GauSAM's source-domain performance on the BCV dataset achieves Dice scores comparable to state-of-the-art methods despite a higher HD, it demonstrates clear advantages in cross-domain scenarios—achieving significantly better Dice scores and lower HD compared to existing approaches. This superior generalization stems from GauSAM's continuous Gaussian probability field, which enables resolution-agnostic predictions and smooth modeling of large, uniform structures like the liver in AMOS dataset, enhancing boundary fidelity. The integration of contour-guided priors via the Non-Subsampled Contourlet Transform (NSCT) and Contour Propagation Unit (CPU) further boosts its robustness across domains. These strengths underscore the versatility and efficiency of GauSAM in medical image segmentation.

H.2 Factors Impacting Multi-Organ Segmentation

The continuous representation that drives GauSAM's strengths also presents challenges for multi organ segmentation on the BCV dataset, influenced by its single feature and probability field design and dataset properties. GauSAM employs a single Gaussian probability field to represent all organ classes, which performs well for large structures but may have limited capacity to fully capture the fine details of smaller organs, given the diversity in size and boundary complexity of the BCV dataset. This design reflects a deliberate trade-off. While adopting multiple class-specific probability fields would significantly increase the number of trainable parameters to approximately 6.6M—substantially raising computational costs and inference latency—we found such overhead unacceptable. In contrast, GauSAM maintains a highly compact architecture with only 2.9M trainable parameters for multi

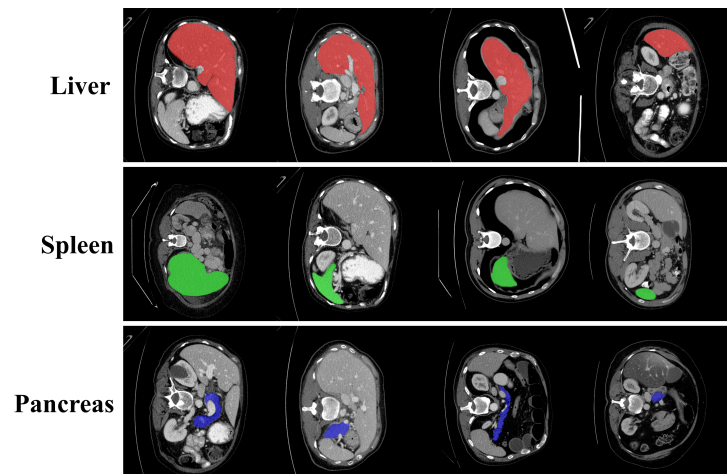


Figure 5: Visualization of 2D axial slices from the BCV dataset, highlighting appearance variability and semantic discontinuity across slices, which impacts consistent learning of organs.

organ segmentation, which is the lowest among models achieving comparable performance. We therefore prioritized efficiency and scalability, accepting a trade-off in fine-grained precision for smaller organs in multi-class settings.

Moreover, to adapt to 2D tasks, we converted the 3D CT volumes in the BCV dataset into axial slices. This slicing process introduces appearance variability across slices and disrupts semantic continuity, particularly affecting consistent learning of smaller organs due to the loss of 3D contextual information essential for volumetric coherence, as illustrated in Figure 5. However, GauSAM currently utilizes 2D Gaussian Splatting to construct continuous feature and probability fields for 2D images—a principle that naturally extends to 3D for reconstructing coherent anatomical structures. As a future direction, this extension can better capture semantic consistency and texture in CT scenarios, enabling more precise modeling while addressing these challenges. The ease of this 2D-to-3D transition further highlights GauSAM’s strong extensibility and generalizability across diverse imaging tasks.

I Broader Impact

I.1 Clinical Impact

GauSAM holds significant promise for advancing medical image segmentation, with direct implications for clinical practice. By improving boundary fidelity and semantic consistency, GauSAM enables more precise structure delineation, facilitating accurate surgical planning and supporting radiologists in diagnosing complex cases. This can lead to earlier disease detection, improved patient outcomes, and reduced workload for medical professionals, particularly in medically underserved settings where automated tools are critical. Furthermore, such advancements may enhance the efficiency of treatment workflows and support telemedicine initiatives, broadening access to high-quality diagnostic tools in underserved regions.

I.2 Resolution-Agnostic Mask Generation

The resolution-agnostic mask generation capability of GauSAM offers substantial practical benefits in medical imaging. Medical imaging modalities, such as CT and MRI, often produce high-resolution images due to the inherent properties of their imaging principles, which capture fine anatomical details. However, performing segmentation directly on these high-resolution images is computationally expensive, requiring significant resources. GauSAM’s ability to generate accurate segmentation masks from downsampled images, while producing masks at the original image resolution with preserved quality, significantly reduces computational demands. This approach enables efficient processing on resource-constrained systems without compromising the precision of boundary delineation in the

final high-resolution output, making it highly applicable in clinical settings where computational efficiency is critical.

I.3 Generalization Beyond Medical Imaging

Although GauSAM was trained and evaluated on medical image segmentation datasets, its design is not exclusively tailored to medical data. The challenges addressed by GauSAM—such as blurred boundaries, insufficient semantic consistency, segmentation artifacts, and inadequate boundary continuity and precision—are prevalent across various segmentation tasks in both medical and non-medical domains. These issues are particularly pronounced in medical image segmentation due to its complexity and significant clinical implications, making it a compelling evaluation setting for validating GauSAM's capabilities. However, the underlying principles of GauSAM, including its continuous Gaussian probability field and contour-guided mechanisms, are broadly applicable to other domains facing similar challenges, such as autonomous driving, remote sensing, or industrial defect detection. By improving segmentation accuracy, reducing artifacts, and enhancing boundary continuity, GauSAM has the potential to deliver strong performance in these fields, underscoring its versatility and robustness across diverse applications.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

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

Answer: [Yes]

Justification: The main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: We discuss the limitations of the work in Section 5

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We have provided sufficient technical, implementation, and experimental details in Section 3 and Section 4, which are adequate to support the reproduction of the main experiments.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: All datasets used in this work are publicly available. The code will be released upon publication of the paper.

Guidelines:

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

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification: All experimental settings and procedures have been described in detail in Section 4.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [\[Yes\]](#)

Justification: In the main experiments, we report error bars for the test results. However, in the other experiments, error bars are not necessary because these evaluations are all conducted using the model that achieved the best Dice score on the validation set in the main experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [\[Yes\]](#)

Justification: The hardware used for training and the estimated training time have already been described in Section 4.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: We conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: We discuss the broader impacts of our work in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [Yes]

Justification: Our work does not have the risk for misuse.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All the assets used in our paper are publicly available for academic research.

Guidelines:

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

13. New assets

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

Answer: [NA]

Justification: Our work does not release new assets.

Guidelines:

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

14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

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

Answer: [NA]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development of our work does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.