

Inter2Former: Dynamic Hybrid Attention for Efficient High-Precision Interactive Segmentation

You Huang, Lichao Chen, Jiayi Ji, Liujuan Cao, Shengchuan Zhang*, Rongrong Ji
Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University

youhuang0607@gmail.com, fpg2012@foxmail.com, jjyxmu@gmail.com
caoliujuan@xmu.edu.cn, zsc_2016@xmu.edu.cn, rrji@xmu.edu.cn

Abstract

Interactive segmentation (IS) improves annotation efficiency by segmenting target regions from user prompts, with widespread applications in real-world scenarios. Current approaches face a critical trade-off: dense-token methods achieve superior accuracy and detail preservation but suffer from prohibitively slow processing on CPU devices, while the Segment Anything Model (SAM) advances the field with sparse prompt tokens for fast inference but compromises segmentation quality. In this paper, we propose Inter2Former to address this challenge by optimizing computation allocation in dense-token processing, which introduces four key enhancements. First, we propose Dynamic Prompt Embedding (DPE) that adaptively processes only regions of interest while avoiding additional overhead from background tokens. Second, we introduce Dynamic Hybrid Attention (DHA), which leverages previous segmentation masks to route tokens through either full attention ($O(N^2)$) for boundary regions or our proposed efficient BSQ attention ($O(N)$) for non-boundary regions. Third, we develop Hybrid Mixture of Experts (HMoE), which applies similar adaptive computation strategies in FFN modules with CPU-optimized parallel processing. Finally, we present Dynamic Local Upsampling (DLU), a reverse operation of DPE, which localizes objects with a lightweight MLP and performs fine-grained upsampling only in detected regions. Experimental results on high-precision IS benchmarks demonstrate that Inter2Former achieves SOTA performance with high efficiency on CPU devices.

1. Introduction

Interactive segmentation (IS) [5, 18, 23, 32] greatly enhances the image segmentation annotation process by segmenting regions of interest from a few annotator prompts,

*Corresponding author

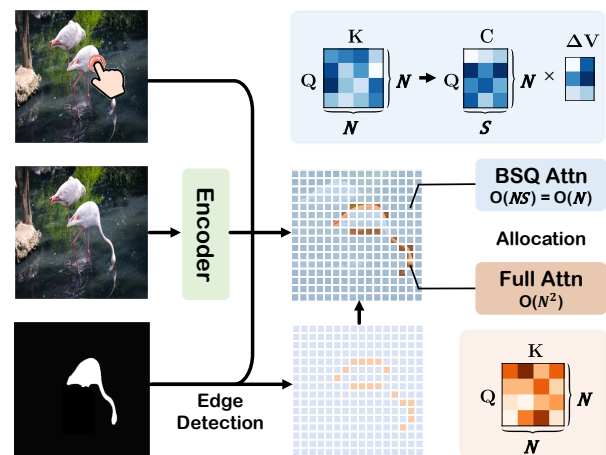


Figure 1. Overview of Dynamic Hybrid Attention (DHA). During interactive segmentation, the model receives user clicks and the previous segmentation mask. DHA leverages mask boundaries to route image tokens through either Full Attention with $O(N^2)$ complexity for boundary regions, or our proposed BSQ Attention with $O(NS) = O(N)$ complexity for non-boundary regions, reducing computation while preserving segmentation quality.

such as clicks [18], bounding boxes [23] or coarse masks [24]. These works expand real-world applications of image segmentation, e.g. medical imaging [2], industrial defect detection [3] and autonomous driving [4]. Recently, the Segment Anything Model (SAM) [23] has become a milestone in IS, excelling in real-time, high-quality segmentation, particularly in mainstream click-based IS.

SAM [23] and InterFormer [18] simultaneously introduce a similar two-stage pipeline for this task: a preprocessing stage that encodes images into tokens using an encoder, followed by an interaction stage where a decoder processes these tokens along with user prompts to generate segmentation masks. While sharing similar encoders, these models differ significantly in their decoder design. InterFormer [18] converts clicks into dense prompt tokens to

enhance spatial awareness and achieve superior segmentation accuracy. However, this approach incurs high computational costs, making it impractically slow on CPU devices, a significant limitation for large-scale crowdsourced annotation with limited GPU resources. In contrast, SAM [23] utilizes sparse prompt tokens for efficient cross-attention and faster inference, but sacrifices spatial awareness and boundary precision as a result. Subsequently, SegNext [34] improves SAM’s decoder by incorporating dense prompt tokens, which enhances accuracy but further increases computational demands. HRSAM [19] focuses on upgrading SAM’s encoder while maintaining its efficient but less accurate sparse-token decoder. Despite these efforts, achieving high-precision interactive segmentation under computational constraints remains challenging [19, 34], especially for high-resolution images where the performance-efficiency trade-off becomes more severe.

To address the aforementioned challenges, we investigate an approach built upon dense prompt tokens [18, 34] while maintaining efficient inference capabilities. We observe that the inefficiency of dense prompt tokens stems from suboptimal resource allocation in the IS process. Specifically, existing models uniformly allocate computation across all tokens during the entire IS process. However, the main object region is typically determined within the first few clicks and most subsequent user clicks focus on refining object boundaries. Computational resources should therefore be prioritized for boundary regions. Thus, this uniform allocation wastes computations on already determined main object regions while providing insufficient computation to critical boundary regions, leading to suboptimal precision and efficiency. Furthermore, each step’s segmentation result in the IS process contains boundary cues, but existing models simply incorporate the segmentation result as an additional feature [18, 23, 24, 32] to refine the next interaction, failing to fully utilize the boundary information for computational optimization.

In this paper, we introduce Inter2Former¹, optimizing computation allocation for efficient high-precision IS. First, we propose Dynamic Prompt Embedding (DPE) that processes only regions of interest while avoiding computational overhead of background tokens through dynamic region cropping. Second, we propose Dynamic Hybrid Attention (DHA) to allocate computational resources differentially between dynamically detected boundary and non-boundary tokens. In DHA (Figure 1), boundary tokens are processed through conventional Full Attention (FA) [45], while non-boundary tokens utilize our lightweight BSQ Attention (BSQA). Our BSQA, inspired by [30], applies Binary Spherical Quantization [54] to compress key-value pairs, reducing complexity from $O(N^2)$ to $O(N)$ for non-

boundary tokens. Third, we propose Hybrid Mixture of Experts (HMoE) in FFN modules to dynamically route boundary and non-boundary tokens to either MoE or conventional FFN for optimized computation allocation. Besides, we optimize MoE computation on CPUs by rearranging tokens to enhance low-level matrix computation and reduce latency. Fourth, we propose Dynamic Local Upsampling (DLU) to speed up mask prediction, functioning as an inverse operation to DPE. DLU first employs a lightweight MLP and Canny operator to localize regions of interest, then performs upsampling exclusively within these regions for fine-grained segmentation at minimal computational cost.

We evaluate Inter2Former on high-precision interactive segmentation benchmarks [19, 21, 34, 40]. Our method achieves SOTA performance with slight additional latency over sparse-token models [19, 23]. The CPU-optimized HMoE reduces inference time by 56-85% over vanilla MoE implementations.

Our main contributions are as follows:

- We propose DHA, which assigns tokens to Full Attention (FA) or our novel BSQ Attention (BSQA) based on prior segmentation results, optimizing computation allocation.
- We propose HMoE for better computation allocation, with optimized parallel MoE computation on CPUs through token rearrangement, reducing inference latency by 56-85% over vanilla MoE implementations.
- We propose DPE and DLU for efficient prompt encoding and mask prediction. DPE dynamically crops regions of interest, while DLU performs the inverse operation by using a lightweight MLP to identify object regions for selective upsampling and fine-grained segmentation.
- Inter2Former achieves SOTA while maintaining high efficiency on CPU devices.

2. Related Work

Interactive Segmentation. Interactive segmentation is first integrated with deep learning in DIOS [50], which introduces deep neural networks to this field and establishes the standard training and evaluation protocol for mainstream click-based interaction. Subsequent research advances this field with improved performance and efficiency [1, 5, 18, 20, 27, 29, 32, 33, 37, 42, 49, 53]. The introduction of SAM [23] marks a significant advancement in inference efficiency through feature reuse, and inspires numerous downstream applications [25, 36, 38]. This evolution leads to the emergence of high-precision interactive segmentation [19, 34, 46], which aims for superior accuracy on precisely annotated datasets [21, 40].

Vector Quantized Representation Learning. The Vector Quantization Variational AutoEncoder (VQ-VAE) [44] pioneered discrete representation learning through learnable codebook quantization. Later works address codebook collapse via perceptual losses [10], ℓ_2 normalization [51], im-

¹Code is available at <https://github.com/YouHuang67/inter2former>.

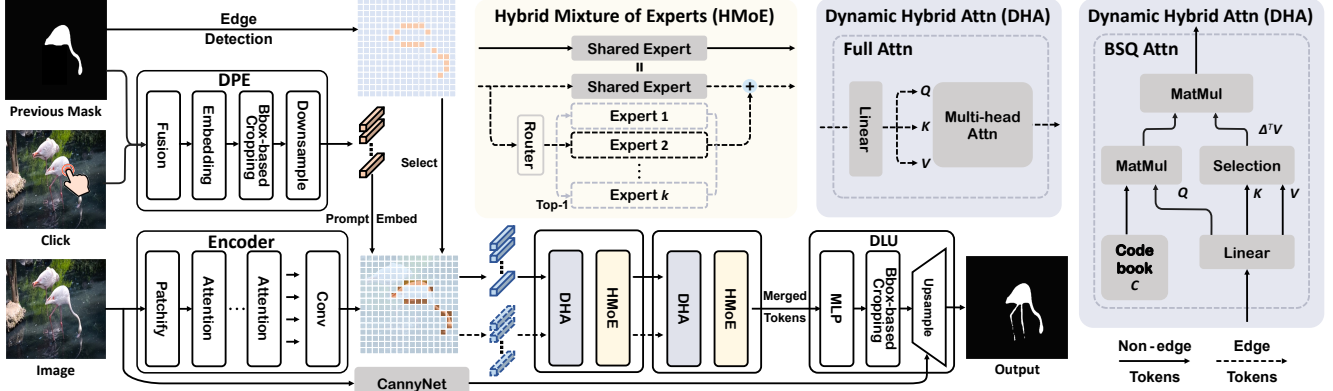


Figure 2. Overview of Inter2Former. Given user clicks and previous mask, our Inter2Former first employ DPE to fuse and transform the prompts into prompt embeddings that are then added to image tokens extracted by the encoder. Based on edge detection from the previous mask, the prompt-fused tokens are divided into edge tokens and non-edge tokens for efficient processing. These tokens undergo DHA and HMoE layers. In DHA, edge tokens are processed by Full Attention while non-edge tokens utilize our proposed BSQA. In HMoE, edge tokens are processed by both multiple routed experts and a shared expert, followed by summation of their outputs, while non-edge tokens only pass through the shared expert. Finally, the merged tokens are upsampled through DLU to generate the output mask.

PLICIT codebooks [39], factorized quantization [9], and dynamic updates [55]. Most notably, BSQ [54] introduces a parameter-free approach by projecting and binarizing latent vectors on a hypersphere, enabling efficient tokenization for vision transformers.

Efficient Attention. Attention [45] has revolutionized computer vision [8, 11, 22, 47]. However, its quadratic computational complexity motivates continuous exploration of efficient solutions. Previous works improve efficiency through local windows [35], sparse patterns [13], and hierarchical structures [15, 26]. At the implementation level, Flash Attention [6, 7] and EMA [41] reduce memory footprint through optimized computation scheduling. Most recently, Transformer-VQ [30] achieves linear time complexity by representing keys through vector quantization, where attention only needs to be computed between queries and a fixed-size codebook rather than the full key sequence.

3. Method

3.1. Background

Interactive segmentation. Interactive segmentation (IS) performs foreground-background segmentation using user prompts. Given an input image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$ and n user clicks $C_n = \{(y_i, x_i, z_i)\}_{i=1}^n$, where $(y_i, x_i) \in [0, H - 1] \times [0, W - 1]$ denotes click coordinates and $z_i \in \{0, 1\}$ indicates positive (foreground) or negative (background) clicks, the IS model outputs a binary mask $\mathbf{M} \in \{0, 1\}^{H \times W}$. While bounding boxes and scribbles are alternative prompts, they can be converted to clicks. Click-based IS has become mainstream due to standardized evaluation and high segmentation accuracy (e.g. IoU > 95%).

Efficient IS pipeline. Traditional IS models convert clicks into click maps, which are two-channel binary masks matching the image size. Each channel stores positive or negative clicks by representing each click as a circular disk, where pixels within radius R (e.g. $R = 5$) from the click center are set to 1 and all other pixels are set to 0. These click maps combine with the input image to form a 5-channel input for vision model inference. However, this process creates computational redundancy through repeated image processing. SAM [23] and InterFormer [18] improve efficiency by decoupling the IS pipeline into two stages: image preprocessing before interaction and lightweight decoding during interaction. In the preprocessing stage, a ViT encoder converts the input image into d -dimensional image tokens $\mathbf{F} = \text{Encoder}(\mathbf{I}) \in \mathbb{R}^{d \times h \times w}$, where $h = H/16, w = W/16$. During interaction, at step k , a lightweight decoder processes these tokens along with current clicks C_k and the previous segmentation mask \mathbf{M}_{k-1} to generate the refined mask $\mathbf{M}_k = \text{Decoder}(\mathbf{F}, C_k, \mathbf{M}_{k-1})$.

Different decoder designs. SAM and InterFormer differ in decoder design: SAM encodes C_n into sparse prompt tokens for efficiency, while InterFormer uses click maps as dense prompt tokens for accuracy. This paper builds upon InterFormer’s architecture and click maps to propose Inter2Former, optimizing dense prompt token computation to achieve SAM-level efficiency.

3.2. Overview of Inter2Former

Encoder. Inter2Former adopts HRSAM’s encoder [19], utilizing the Flash Swin and multi-scale fusion to generate image tokens $\mathbf{F} \in \mathbb{R}^{d \times h \times w}$. Considering that HRSAM’s Selective State Space (SSM) operator [12] cannot be com-

puted on CPU, we simplify the HRSAM encoder by removing SSM while maintaining its visual extrapolation capability [19] for high-resolution images.

Decoder. As illustrated in Figure 2, the Inter2Former decoder comprises three main components. First, following InterFormer’s prompt embedding scheme [18], user clicks are encoded into click maps and combined with the previous mask to generate prompt tokens $\mathbf{P} \in \mathbb{R}^{d \times h \times w}$, which are dimensionally aligned with image tokens \mathbf{F} via our efficient DPE that focuses on local regions. Second, adopting the standard transformer architecture [45], we process the prompt-fused tokens $\mathbf{F}_P = \mathbf{F} + \mathbf{P}$ through alternating layers of our DHA (Attention) and HMoE (FFN) modules to obtain refined tokens $\mathbf{F}_R \in \mathbb{R}^{d \times h \times w}$. Finally, we leverage DLU to upsample \mathbf{F}_R into the final segmentation mask.

3.3. Dynamic Prompt Embedding

Following InterFormer [18], we maintain a reference mask $\mathbf{M}_{\text{ref}} \in \{0, 1, 2, 3, 4\}^{H \times W}$ to encode user clicks and previous predictions. In \mathbf{M}_{ref} , values 0 and 4 mark definite background and foreground regions within radius-5 circles of clicks. Values 1 and 3 indicate possible background and foreground from predictions. Value 2 represents uncertain regions. During interaction, we update \mathbf{M}_{ref} using InterFormer’s rules based on new clicks and predictions.

To reduce computational cost, we propose Dynamic Prompt Embedding (DPE) for efficient \mathbf{M}_{ref} encoding. We first detect a bounding box $\mathcal{B} = [x_1 : x_2, y_1 : y_2]$ containing all click regions and foreground predictions with padding. For the local region of \mathcal{B} , we apply learnable embedding:

$$\mathbf{E}_{\text{init}} = \text{Embed}(\mathbf{M}_{\text{ref}}[y_1 : y_2, x_1 : x_2]) \in \mathbb{R}^{5 \times H_{\mathcal{B}} \times W_{\mathcal{B}}}, \quad (1)$$

where $H_{\mathcal{B}} = y_2 - y_1$, $W_{\mathcal{B}} = x_2 - x_1$, and $\text{Embed}(\cdot)$ maps each value to a learnable 5-dimensional vector. The features are downsampled through four stride-2 convolutions:

$$\mathbf{F}_{\mathcal{B}} = \text{Conv}_4(\mathbf{E}_{\text{init}}) \in \mathbb{R}^{d \times h_{\mathcal{B}} \times w_{\mathcal{B}}}, \quad (2)$$

where $h_{\mathcal{B}} = H_{\mathcal{B}}/16$ and $w_{\mathcal{B}} = W_{\mathcal{B}}/16$. The final prompt embedding combines local features with a learnable background embedding \mathbf{e}_{bg} :

$$\mathbf{P} = \begin{bmatrix} \mathbf{e}_{\text{bg}} & \cdots & \mathbf{e}_{\text{bg}} \\ \vdots & \mathbf{F}_{\mathcal{B}} & \vdots \\ \mathbf{e}_{\text{bg}} & \cdots & \mathbf{e}_{\text{bg}} \end{bmatrix} \in \mathbb{R}^{d \times h \times w}. \quad (3)$$

This design reduces computation by processing only regions of interest while maintaining global context through background embedding.

3.4. Dynamic Hybrid Attention

Computation Allocation. To efficiently allocate computation at interaction step k , we identify object boundaries from the previous mask \mathbf{M}_{k-1} . We first apply a 7×7 average convolution (uniform weights $1/49$) to estimate lo-

cal variance and detect non-zero variance regions. The resulting edge map is then downsampled via max pooling to match the dimensions of \mathbf{F}_P :

$$\mathbf{E}_{k-1} = \text{Pool}(\mathbb{1}\{\text{Conv}(\mathbf{M}_{k-1}^2) - \text{Conv}(\mathbf{M}_{k-1})^2 > 0\}). \quad (4)$$

Guided by the edge map \mathbf{E}_{k-1} , we allocate computational resources as follows. The prompt-fused tokens \mathbf{F}_P are reshaped into sequence form $\mathbf{F}_{\text{flat}} = \text{Reshape}(\mathbf{F}_P) \in \mathbb{R}^{L \times d}$, where $L = h \times w$. Linear projections are then applied to generate query, key and value matrices: $(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\mathbf{F}_{\text{flat}} \mathbf{W}_q, \mathbf{F}_{\text{flat}} \mathbf{W}_k, \mathbf{F}_{\text{flat}} \mathbf{W}_v) \in \mathbb{R}^{L \times C}$. Based on \mathbf{E}_{k-1} , we partition queries into two disjoint sets:

$$\begin{aligned} \mathbf{Q}_{\text{FA}} &= \{\mathbf{q}_i | \mathbf{E}_{k-1}(i) = 1\} \\ \mathbf{Q}_{\text{BSQ}} &= \{\mathbf{q}_i | \mathbf{E}_{k-1}(i) = 0\} \end{aligned} \quad (5)$$

where \mathbf{Q}_{FA} corresponds to edge regions (minority) and \mathbf{Q}_{BSQ} to non-edge regions (majority). Both sets share the same key-value pairs (\mathbf{K}, \mathbf{V}) but employ different attention computations: \mathbf{Q}_{FA} uses full attention with $O(N^2)$ complexity, while \mathbf{Q}_{BSQ} adopts BSQA with $O(N)$ complexity, thus balancing effectiveness and efficiency.

Full Attention. For queries in \mathbf{Q}_{FA} , we apply the standard full attention computation. Given the shared key-value pairs $(\mathbf{K}, \mathbf{V}) \in \mathbb{R}^{N \times C}$, where C is the head dimension (we present the single-head case for clarity, while our implementation adopts standard multi-head attention as in transformers), the attention output is computed as:

$$\text{FA}(\mathbf{Q}_{\text{FA}}) = \text{Softmax}\left(\frac{\mathbf{Q}_{\text{FA}} \mathbf{K}^{\top}}{\sqrt{C}}\right) \mathbf{V}. \quad (6)$$

This FA computation has quadratic complexity $O(N^2)$ with respect to sequence length. For the majority of queries in \mathbf{Q}_{BSQ} , we introduce our proposed BSQA that achieves linear complexity while maintaining attention effectiveness.

3.5. Binary Spherical Quantization Attention

Inspired by VQ-Transformer [30], we propose BSQA that enhances VQ Attention with a more effective Binary Spherical Quantization (BSQ) scheme while maintaining its linear-time complexity. Below we elaborate on VQ Attention, BSQ and our proposed BSQA respectively.

Vector Quantized Attention. The insight of VQ Attention is applying vector quantization to the N key vectors $\mathbf{K} \in \mathbb{R}^{N \times C}$ through a learned codebook $\mathbf{C} \in \mathbb{R}^{S \times C}$ of size S . Each key vector is mapped to its nearest codebook vector:

$$\hat{\mathbf{K}} = \text{VQ}(\mathbf{K}, \mathbf{C}) = \arg \min_{\mathbf{c}_i \in \mathbf{C}} \|\mathbf{k} - \mathbf{c}_i\|_2^2. \quad (7)$$

This quantization process can be expressed through a binary assignment matrix $\Delta \in \{0, 1\}^{N \times S}$, where $\hat{\mathbf{K}} = \Delta \mathbf{C}$. The attention computation (without softmax’s normalization) can then be reformulated as:

$$\exp(\mathbf{Q} \hat{\mathbf{K}}^{\top}) \mathbf{V} = \exp(\mathbf{Q} \mathbf{C}^{\top}) (\Delta^{\top} \mathbf{V}). \quad (8)$$

The binary nature of Δ enables this equivalent factorization, which reduces the computational complexity from quadratic $O(N^2)$ to linear $O(NS) = O(N)$ with fixed S . The complete attention with softmax normalization can be computed by applying the same factorization twice: once with the \mathbf{V} , and once with an all-ones vector $\mathbf{1}_{N \times 1}$:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \frac{\exp(\mathbf{Q}\mathbf{C}^\top)(\Delta^\top \mathbf{V})}{\exp(\mathbf{Q}\mathbf{C}^\top)(\Delta^\top \mathbf{1}_{N \times 1})}. \quad (9)$$

However, nearest-neighbor quantization $\text{VQ}(\cdot, \cdot)$ tends to underutilize the codebook by activating only a small subset of vectors. More critically, when applying straight-through estimation (STE), the gradient approximation relies on copying gradients from quantized $\hat{\mathbf{K}}$ to \mathbf{K} :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{K}} \approx \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{K}}}. \quad (10)$$

Although minimizing $\|\hat{\mathbf{K}} - \mathbf{K}\|_2^2$ is introduced as a commitment loss to bound this approximation, the quantization error remains uncontrollable during training. To address these fundamental limitations, we adopt BSQ instead.

Binary Spherical Quantization. BSQ [54] addresses the limitations of VQ by projecting features onto a unit hypersphere before quantization. Likewise, given N key vectors $\mathbf{K} \in \mathbb{R}^{N \times C}$, BSQ first maps it to a lower-dimensional space through a learnable transformation:

$$\mathbf{B} = \mathbf{K}\mathbf{W}_{\text{BSQ}} \in \mathbb{R}^{N \times S} \text{ where } S \ll C. \quad (11)$$

The projected vector is then normalized onto a unit sphere:

$$\mathbf{U} = \mathbf{B} / \|\mathbf{B}\|_2. \quad (12)$$

Unlike VQ's nearest-neighbor assignment, BSQ applies element-wise binary quantization:

$$\hat{\mathbf{U}} = \text{sign}(\mathbf{U}) / \sqrt{S}, \quad (13)$$

where the scaling factor ensures the quantized vector maintains unit norm. The quantized features are then projected back to the original dimension through another learnable transformation, which will be discussed in the next section. Since BSQ's quantization error $\|\mathbf{U} - \hat{\mathbf{U}}\|$ is theoretically bounded [54] and empirically converges close to zero during our experiments, it enables accurate gradient estimation through STE, addressing VQ's gradient approximation challenge. Next, we will introduce BSQA that addresses the inefficient codebook utilization issue in VQ Attention.

BSQ Attention. To leverage $\hat{\mathbf{U}}$ in attention computation similar to VQ Attention's codebook vectors, we first transform $\hat{\mathbf{U}}$ through a simple linear mapping: $\mathbf{I} = \hat{\mathbf{U}} \times \sqrt{S}/2 + 1/2 \in \{0, 1\}^{N \times S}$, which encodes each key as an S -bit binary representation. We then construct the final projected vectors by combining learnable base vectors $\mathbf{C}_{\text{base}}^0, \mathbf{C}_{\text{base}}^1 \in \mathbb{R}^{S \times C}$. Specifically, for each bit position j , we select either $\mathbf{C}_{\text{base}}^0$ or $\mathbf{C}_{\text{base}}^1$ based on the binary value, and aggregate these selections across all bit positions. This aggregation

process can be efficiently formulated as

$$\hat{\mathbf{K}} = [\mathbf{I} \quad \mathbf{1} - \mathbf{I}] \begin{bmatrix} \mathbf{C}_{\text{base}}^1 \\ \mathbf{C}_{\text{base}}^0 \end{bmatrix} \in \mathbb{R}^{N \times C}. \quad (14)$$

The S -bit binary representation allows 2^S different combinations $\{0, 1, \dots, 2^S - 1\}$. Through our binary selection mechanism with $\mathbf{C}_{\text{base}}^0$ and $\mathbf{C}_{\text{base}}^1$, we naturally construct a codebook of 2^S vectors. Following the formulation similar to Eq. 14, these vectors constitute the codebook required for VQ Attention, enabling computations analogous to Eq. 8.

3.6. Hybrid Mixture of Experts

HMoE builds upon DeepSeek V3's MoE design [31] including its auxiliary-loss-free expert balancing strategy. Like DHA, HMoE adopts a hybrid strategy where the computation path for each token is determined by the edge map \mathbf{E}_{k-1} . With M routed experts $\{\text{FFN}_i\}_{i=0}^{M-1}$ and one shared expert FFN_M , for input tokens $\mathbf{X} \in \mathbb{R}^{N \times d}$, non-edge tokens where $\mathbf{E}_{k-1}(t) = 0$ only utilize the shared expert:

$$\mathbf{y}_t = \text{FFN}_M(\mathbf{x}_t). \quad (15)$$

For edge tokens where $\mathbf{E}_{k-1}(t) = 1$, we first compute token-expert affinity scores:

$$s_{i,t} = \text{Sigmoid}(\mathbf{x}_t^\top \mathbf{e}_i), 0 \leq i \leq M, \quad (16)$$

where \mathbf{e}_i represents the learnable centroid vector of the i -th expert. The token then selects the expert with highest affinity among M routed experts:

$$a_t = \arg \max_{0 \leq i \leq M-1} s_{i,t}, \quad (17)$$

and combines its output with the shared expert:

$$\mathbf{y}_t = \frac{\exp(s_{M,t})\text{FFN}_M(\mathbf{x}_t) + \exp(s_{a_t,t})\text{FFN}_{a_t}(\mathbf{x}_t)}{\exp(s_{M,t}) + \exp(s_{a_t,t})}. \quad (18)$$

Efficient Parallel Processing. To enable efficient matrix operations, we propose a token rearrangement strategy specifically for edge tokens. Using their selected expert indices $\{a_t | \mathbf{E}_{k-1}(t) = 1\}$ where $a_t \in \{0, 1, \dots, M-1\}$, we first sort these tokens to obtain contiguous groups:

$$\pi = \text{argsort}(\{a_t | \mathbf{E}_{k-1}(t) = 1\}), \quad (19)$$

where π is the permutation that sorts edge tokens by expert ID. The sorted tokens form M groups:

$$\mathbf{X}_i = \{\mathbf{x}_{\pi[j]} | a_{\pi[j]} = i\} \in \mathbb{R}^{N_i \times d}, \quad (20)$$

where N_i is the number of edge tokens assigned to routed expert i . By rearranging tokens assigned to the same expert into contiguous memory blocks, we can decompose each expert's FFN computation into matrix multiplications (as Linear layers are essentially matrix multiplications) with activation functions. Through C++ extension with low-level matrix operation optimization, these blocked matrix operations for multiple experts are efficiently executed simultaneously. The final outputs are recovered by inverse per-

mutation π^{-1} . This multi-expert batch processing scheme achieves significant speedup over sequential computation.

3.7. Dynamic Local Upsampling

Local Refinement. We apply DLU to generate the segmentation mask from tokens $\mathbf{F}_R \in \mathbb{R}^{d \times h \times w}$ refined by DHA and HMoE modules. Our DLU consists of a localization branch and a refinement branch for efficient computation allocation. The localization branch first generates a coarse mask without upsampling to identify the object region:

$$\mathbf{M}_{\text{low-res}} = \text{MLP}(\mathbf{F}_R; d \rightarrow 1) \in \mathbb{R}^{1 \times h \times w}, \quad (21)$$

from which we extract an expanded bounding box $\mathcal{B} = [x_1 : x_2, y_1 : y_2]$ around detected objects. The refinement branch then focuses on precise boundary delineation within the detected region. We first extract tokens within the expanded bounding box \mathcal{B} , where $h_{\mathcal{B}} = y_2 - y_1$ and $w_{\mathcal{B}} = x_2 - x_1$ define the local region dimensions:

$$\mathbf{F}_{\mathcal{B}} = \mathbf{F}_R[:, y_1 : y_2, x_1 : x_2] \in \mathbb{R}^{d \times h_{\mathcal{B}} \times w_{\mathcal{B}}}. \quad (22)$$

The refinement branch leverages edge features to guide the upsampling of $\mathbf{F}_{\mathcal{B}}$ for precise boundary delineation.

Edge-guided Upsampling. We extract multi-scale edge features through a small CannyNet prior to interaction:

$$\begin{aligned} \mathbf{E} &= \text{Canny}(\mathbf{I}), \\ \{\mathbf{F}_i^e\}_{i=1}^4 &= \text{ResNet}(\mathbf{E}), \end{aligned} \quad (23)$$

where \mathbf{E} denotes the binary edge map and $\{\mathbf{F}_i^e\}_{i=1}^4$ represents edge features at increasing scales. $\{\mathbf{F}_i^e\}_{i=1}^4$ are fused with $\mathbf{F}_{\mathcal{B}}$ during upsampling to generate the final mask:

$$\mathbf{M} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \text{Up}(\mathbf{F}_{\mathcal{B}}, \{\mathbf{F}_i^e\}_{i=1}^4) & \vdots \\ 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{H \times W}, \quad (24)$$

where $\text{Up}(\cdot)$ consists of four deconvolution layers ($\times 2$) that progressively expand resolution while reducing channels to 1. At each upsampling stage, the corresponding \mathbf{F}_i^e is fused with the upsampled feature map via addition, followed by a convolutional layer for feature refinement.

3.8. Training Strategy

BSQA Training. During training, we replace keys in attention computation using Eq. 14 while performing complete FA computation, only with the quantized key vectors. This training strategy encourages the quantized attention to closely approximate the behavior of standard attention. At inference time, we switch to the efficient computation scheme in Eq. 8 to achieve linear-time complexity.

DLU Training. The DLU module generates two outputs during training: a low-resolution mask $\mathbf{M}_{\text{low-res}}$ from Eq. 21 and a high-resolution mask \mathbf{M} by applying $\text{Up}(\cdot)$ from Eq. 22 to the full token set. These outputs are supervised by downsampled and original GT masks respectively, using

the NFL loss [24] common in interactive segmentation. At inference, we employ DLU for efficient mask generation.

4. Experiments

Section 4.1 details our implementation. Section 4.2 describes datasets and training protocols. Section 4.3 presents the main quantitative comparison. Section 4.4 analyzes computational efficiency of our key components. Additionally, we provide the `code` in the supplementary materials.

4.1. Implementation Details

Our Inter2Former adopts the encoder from HRSAM++ [19], specifically a ViT-Base with 12 layers, 768 embedding dimensions, and 12 attention heads, initialized with MAE pre-training [17]. For CPU compatibility, we remove the Cycle-scan module (with SSM [12]) from the original HRSAM++ while retaining its anchor map (fixed size of 512). For fair comparison, we also re-train HRSAM++ without SSM under the same settings.

The decoder of Inter2Former consists of two alternating layers of DHA and HMoE modules. While it processes 256-dimensional input tokens, the attention computation is performed with reduced 32-dimensional Q/K/V vectors for efficiency. We enhance FA with Rotary Position Embedding (RoPE) [43] and implement BSQA using an 8-bit codebook (16 base vectors). The DPE module transforms the original-scale input through four convolutional layers into 256-dimensional prompt tokens at 1/16 resolution, while DLU reverses this process through four convolutions to generate single-channel full-resolution masks. The proposed CannelNet adopts a straightforward design with four stages of dual ResNet blocks [16], processing features of dimensions 4, 16, 64 and 256 respectively, with each stage halving the spatial resolution.

4.2. Experimental Setting

Datasets. Following recent high-precision interactive segmentation benchmarks [19, 34], we train our model on COCO [28], LVIS [14] and HQSeg-44K [21] datasets. We evaluate on HQSeg-44K validation set and DAVIS [40], both of which contain high-quality annotated masks for rigorous evaluation over high-precision IS tasks.

Training. Following previous high-precision IS works, we adopt two training protocols for comprehensive evaluation. The first protocol follows [34], where we train on COCO and LVIS for 160K iterations, followed by 40K iterations of fine-tuning on HQSeg44K. The second protocol involves distillation from SAM-ViT-Huge [21] where we first train the encoder for 160K iterations on unlabeled images from COCO and LVIS using MSE loss to align with SAM features, then randomly initialize the Inter2Former decoder and train the complete model for 80K iterations on HQSeg44K. Both protocols utilize click simulations from

Model	Training Data	CPU Time (ms) ↓ ‡20-SPC / Online	HQSeg44K _{Max H/W>4000}			DAVIS _{Max H/W<1000}		
			5-mIoU ↑	NoC90 ↓	NoC95 ↓	5-mIoU ↑	NoC90 ↓	NoC95 ↓
RITM-HRNet32 ₄₀₀ [24]	COCO+LVIS	277 / 277	77.72	10.01	14.58	89.75	5.34	11.45
FocalClick-SegF-B3-S2 ₂₅₆ [5]	COCO+LVIS	97 / 97	84.63	8.12	12.63	90.82	5.17	11.42
FocalClick-SegF-B3-S2 ₃₈₄ [5]	COCO+LVIS	175 / 175	85.45	7.03	10.74	91.22	4.90	10.40
SimpleClick-ViT-B ₄₄₈ [32]	COCO+LVIS	212 / 212	85.11	7.47	12.39	90.73	5.06	10.37
InterFormer-ViT-B ₁₀₂₄ [18]	COCO+LVIS	1020 / 188	82.62	7.17	10.77	87.79	5.45	11.88
SegNext (SA×1) ViT-B ₁₀₂₄ [34]	COCO+LVIS	910 / 803	85.41	7.47	11.94	90.13	5.46	13.31
SegNext (SA×2) ViT-B ₁₀₂₄ [34]	COCO+LVIS	1519 / 1400	85.71	7.18	11.52	89.85	5.34	12.80
SegNext (SA×2) ViT-B ₁₀₂₄ [34]	COCO+LVIS+HQ	1519 / 1400	91.75	5.32	9.42	91.87	4.43	10.73
HRSAM ⁺⁺ -ViT-B ₁₀₂₄ [19]	COCO+LVIS+HQ	65 / 40	90.32	6.27	10.14	90.40	5.71	12.72
HRSAM ⁺⁺ -ViT-B ₂₀₄₈ [19]	COCO+LVIS+HQ	273 / 105	91.50	5.41	9.08	90.79	5.52	10.84
SAM-ViT-B ₁₀₂₄ [23]	SA-1B	142 / 40	86.16	7.46	12.42	90.95	5.14	10.74
MobileSAM-ViT-T ₁₀₂₄ [52]	SA-1B	35 / 24	81.98	8.70	13.83	89.18	5.83	12.74
EfficientSAM-ViT-T ₁₀₂₄ [48]	ImageNet+SA-1B	88 / 27	77.90	10.11	14.60	85.26	7.37	14.28
EfficientSAM-ViT-T ₂₀₄₈ [48]	ImageNet+SA-1B	706 / 77	74.20	9.47	13.13	84.10	8.00	14.37
EfficientSAM-ViT-S ₁₀₂₄ [48]	ImageNet+SA-1B	127 / 23	79.01	8.84	13.18	87.55	6.37	12.26
EfficientSAM-ViT-S ₂₀₄₈ [48]	ImageNet+SA-1B	1658 / 69	74.91	8.27	11.97	85.17	6.86	12.49
HQ-SAM-ViT-B ₁₀₂₄ [36]	SA-1B+HQ	167 / 54	89.85	6.49	10.79	91.77	5.26	10.00
HRSAM ⁺⁺ -ViT-B ₁₀₂₄ [19]	†SA-1B+HQ	65 / 40	89.37	6.56	10.61	86.66	7.29	14.32
HRSAM ⁺⁺ -ViT-B ₂₀₄₈ [19]	†SA-1B+HQ	273 / 105	90.94	5.86	9.22	88.46	6.61	12.39
Inter2Former-ViT-B ₁₀₂₄ (Ours)	COCO+LVIS+HQ	75 / 50	91.48	5.36	9.29	90.82	4.90	11.33
Inter2Former-ViT-B ₂₀₄₈ (Ours)	COCO+LVIS+HQ	300 / 131	92.28	4.58	7.79	91.30	4.33	8.45
Inter2Former-ViT-B ₁₀₂₄ (Ours)	†SA-1B+HQ	75 / 50	91.32	5.46	9.34	90.36	4.96	12.72
Inter2Former-ViT-B ₂₀₄₈ (Ours)	†SA-1B+HQ	300 / 131	92.68	4.24	7.39	92.00	4.29	7.82

Table 1. Performance evaluation over high-precision IS tasks. We compare our Inter2Former with SOTA methods on HQSeg44K and DAVIS datasets under two training protocols: (1) conventional COCO+LVIS training with HQ fine-tuning, and (2) distillation from SAM followed by HQ training († indicates models trained via this protocol). For efficiency, we report CPU inference time (‡ where 20-SPC evaluates the average time for standard 20 clicks including preprocessing, *e.g.* SAM’s image encoding, while Online measures per-click latency during interaction excluding preprocessing). While Inter2Former shows marginally higher latency compared to sparse-token-based HRSAM, it demonstrates superior segmentation quality and achieves SOTA performance across 5-mIoU and NoC@90/95.

prior works [18, 24, 32] and NFL [24] to guide the model in producing GT-aligned segmentation masks from clicks.

Evaluation. Following standard evaluation protocols [5, 18, 23, 24, 32, 34, 48, 52], the segmentation quality is measured by NoC@90/95 (average number of clicks required to achieve 90% or 95% IoU within a 20-click budget) and 5-mIoU (mIoU after five clicks) in high-precision IS tasks [19, 34]. For efficiency evaluation on CPU devices, we employ two metrics: 20-SPC (Seconds Per Click) [19] and Online SPC. The 20-SPC metric measures the average time to complete 20 clicks, including preprocessing overhead (*e.g.*, image encoding in SAM-based methods). In contrast, Online SPC captures the per-click latency during interaction, excluding preprocessing time. For traditional IS models preceding SAM, these two metrics yield identical values since they do not require preprocessing steps.

4.3. Main Results

As shown in Table 1, Inter2Former achieves SOTA performance across all evaluation metrics while maintaining competitive efficiency. The superior performance can be attributed to the dense-token design in Inter2Former’s decoder. Unlike HRSAM⁺⁺ which inherits SAM’s sparse-token decoder and struggles with limited training data,

Inter2Former’s dense-token design enables more effective training from scratch, particularly beneficial when fine-tuning on high-quality datasets. In terms of efficiency, while Inter2Former shows marginally higher latency compared to HRSAM⁺⁺, it maintains substantial speed advantages over other high-precision methods like SegNext while achieving better accuracy. Furthermore, Inter2Former demonstrates competitive efficiency against fast SAM variants like EfficientSAM [48], and significantly reduces the inference time compared to InterFormer [18].

4.4. Efficiency Analysis

We evaluate the computational efficiency of our proposed DPE, DHA and DLU on COCO dataset using object mask annotations. The diverse object scales provide objective benchmarks for efficiency analysis under various area ratios (A_{obj}/A_{img}). Additionally, we analyze HMoE’s efficiency across different numbers of experts.

DPE Efficiency. As shown in Figure 3 (a), Non-DPE (with identical convolutional architecture) maintains a constant latency regardless of object size, while our DPE shows a linear increase with area ratios but maintains significant efficiency advantages. Notably, for predominant small objects, DPE requires < 25% of the Non-DPE latency.

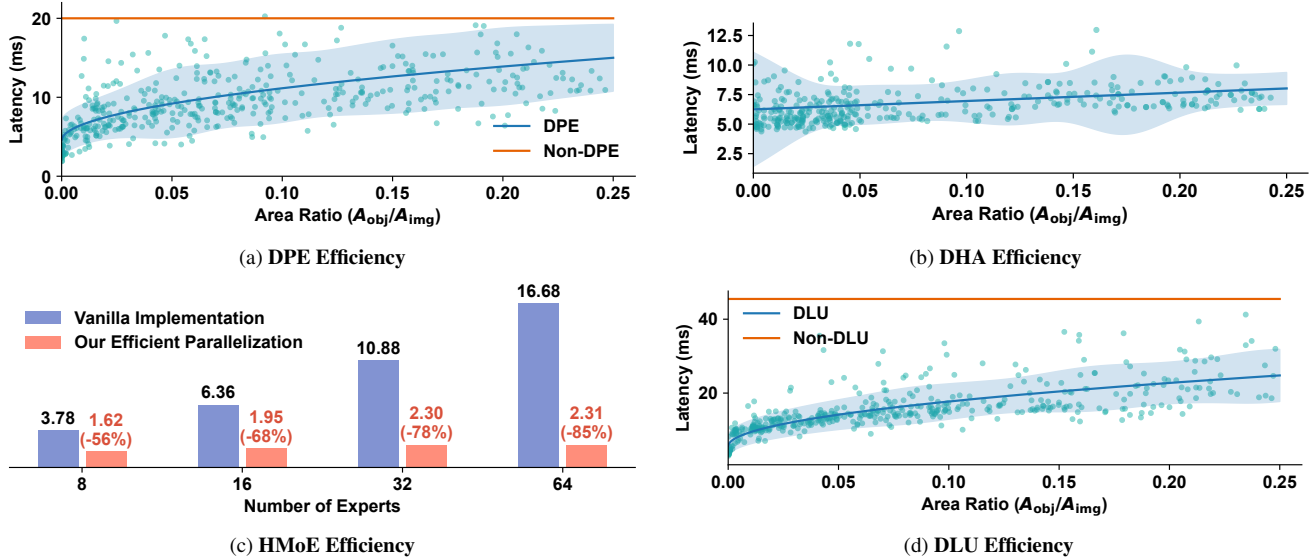


Figure 3. Efficiency analysis. (a) Latency of DPE and Non-DPE (full prompt embedding). (b) DHA latency across different area ratios. (c) HMoE speedup with various expert numbers. (d) Latency of DLU and Non-DLU (full upsampling).

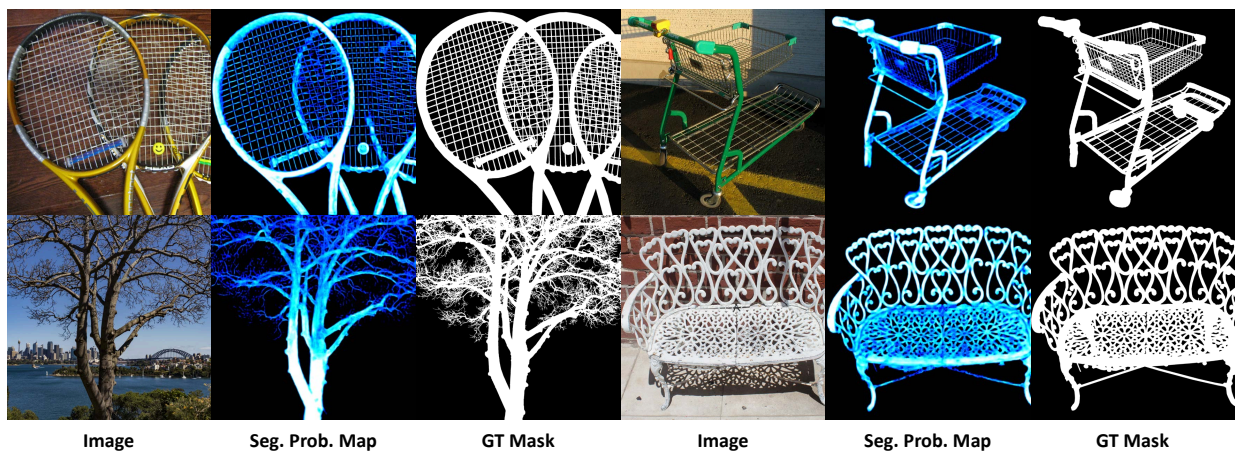


Figure 4. Qualitative results. Inter2Former produces precise segmentation maps for challenging line structures under 20-click prompts.

DHA Efficiency. As shown in Figure 3 (b), DHA shows a slow linear rise in latency as area ratios grow. This stable trend comes from DHA’s focus on edges. While object area grows quadratically, the number of edge pixels grows more slowly, leading to efficient processing even for large objects.

HMoE Efficiency. As shown in Figure 3 (c), we compare our efficient parallel HMoE with vanilla implementation across expert counts. HMoE shows clear speedup with more experts, reaching 85% latency reduction at 64 experts.

DLU Efficiency. As shown in Figure 3 (d), DLU as the inverse of DPE shows similar speed patterns. Our DLU runs much faster than Non-DLU with the same structure.

Detailed ablation studies for each module’s performance are provided in the **supplementary material**.

4.5. Qualitative Results

As shown in Figure 4, Inter2Former achieves precise segmentation on challenging thin structures under 20 clicks.

5. Conclusion

This paper introduces Inter2Former for high-precision interactive segmentation, addressing the critical trade-off between performance and efficiency in dense-prompt-token processing. Through adaptive computation allocation implemented by the proposed Dynamic Prompt Embedding, Dynamic Hybrid Attention, Hybrid Mixture of Experts and Dynamic Local Upsampling, Inter2Former optimizes performance while maintaining efficiency, achieving SOTA performance with competitive speed on CPU devices.

Acknowledgments

This work was supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. U23A20383, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No. 2021J06003, No.2022J06001).

References

- [1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 859–868. IEEE Computer Society, 2018. 2
- [2] Saber Mirzaee Bafti, Chee Siang Ang, Md. Moinul Hossain, Gianluca Marcellini, Marc Alemany-Fornes, and Anastasios D. Tsoulos. A crowdsourcing semi-automatic image segmentation platform for cell biology. 2021. 1
- [3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1
- [4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1
- [5] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: towards practical interactive image segmentation. pages 1300–1309, 2022. 1, 2, 7
- [6] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023. 3
- [7] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022. 3
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3
- [9] Alaaeldin El-Nouby, Matthew J Muckley, Karen Ullrich, Ivan Laptev, Jakob Verbeek, and Hervé Jégou. Image compression with product quantized masked image modeling. *arXiv preprint arXiv:2212.07372*, 2022. 3
- [10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2
- [11] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 3
- [12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3, 6
- [13] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In *Proceedings of the 2019 Conference of the North*, 2019. 3
- [14] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019. 6
- [15] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5961–5971, 2023. 3
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6
- [17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6
- [18] You Huang, Hao Yang, Ke Sun, Shengchuan Zhang, Lijuan Cao, Guannan Jiang, and Rongrong Ji. Interformer: Real-time interactive image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22301–22311, 2023. 1, 2, 3, 4, 7
- [19] You Huang, Wenbin Lai, Jiayi Ji, Lijuan Cao, Shengchuan Zhang, and Rongrong Ji. Hrsam: Efficiently segment anything in high-resolution images. *arXiv preprint arXiv:2407.02109*, 2024. 2, 3, 4, 6, 7
- [20] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5297–5306. Computer Vision Foundation / IEEE, 2019. 2
- [21] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. 2, 6
- [22] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 3
- [23] 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 any-

- thing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2, 3, 7
- [24] Anton Konushin Konstantin Sofiiuk, Ilia A. Petrov. Reviving iterative training with mask guidance for interactive segmentation. *arXiv: Computer Vision and Pattern Recognition*, 2021. 1, 2, 6, 7
- [25] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2
- [26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. 3
- [27] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 577–585. IEEE Computer Society, 2018. 2
- [28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, 2014. 6
- [29] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13336–13345. IEEE, 2020. 2
- [30] Lucas D Lingle. Transformer-vq: Linear-time transformers via vector quantization. *arXiv preprint arXiv:2309.16354*, 2023. 2, 3, 4
- [31] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 5
- [32] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. *arXiv preprint arXiv:2210.11006*, 2022. 1, 2, 7
- [33] Qin Liu, Meng Zheng, Benjamin Planche, Srikrishna Karanam, Terrence Chen, Marc Niethammer, and Ziyang Wu. Pseudoclick: Interactive image segmentation with click imitation. pages 728–745, 2022. 2
- [34] Qin Liu, Jaemin Cho, Mohit Bansal, and Marc Niethammer. Rethinking interactive image segmentation with low latency, high quality, and diverse prompts. *arXiv preprint arXiv:2404.00741*, 2024. 2, 6, 7
- [35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3
- [36] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 2, 7
- [37] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 616–625. IEEE Computer Society, 2018. 2
- [38] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023. 2
- [39] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. 3
- [40] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 724–732. IEEE Computer Society, 2016. 2, 6
- [41] Markus N. Rabe and Charles Staats. Self-attention does not need $o(n^2)$ memory. *CoRR*, abs/2112.05682, 2021. 3
- [42] Konstantin Sofiiuk, Ilia A. Petrov, Olga Barinova, and Anton Konushin. F-BRS: rethinking backpropagating refinement for interactive segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8620–8629. IEEE, 2020. 2
- [43] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 6
- [44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 2, 3, 4
- [46] Bin Wang, Anwesa Choudhuri, Meng Zheng, Zhongpai Gao, Benjamin Planche, Andong Deng, Qin Liu, Terrence Chen, Ulas Bagci, and Ziyang Wu. Order-aware interactive segmentation. *arXiv preprint arXiv:2410.12214*, 2024. 2
- [47] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023. 3
- [48] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xi-ang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. *arXiv preprint arXiv:2312.00863*, 2023. 7
- [49] Long Xu, Yongquan Chen, Rui Huang, Feng Wu, and Shiwu Lai. Structured click control in transformer-based interactive segmentation. *arXiv preprint arXiv:2405.04009*, 2024. 2

- [50] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 373–381. IEEE Computer Society, 2016. [2](#)
- [51] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. [2](#)
- [52] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. [7](#)
- [53] Tiezheng Zhang, Xiaoxi Chen, Chongyu Qu, Alan Yuille, and Zongwei Zhou. Leveraging ai predicted and expert revised annotations in interactive segmentation: Continual tuning or full training? *arXiv preprint arXiv:2402.19423*, 2024. [2](#)
- [54] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024. [2](#), [3](#), [5](#)
- [55] Chuanxia Zheng and Andrea Vedaldi. Online clustered code-book. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22798–22807, 2023. [3](#)