HPT-SAM2: Adapting Segment Anything Model 2 for Camouflaged Object Detection via Hierarchical Prototypes and Texture Modulation

Yu Chen[1], Chengliang Wang[1,*], Xing Wu[1], Peng Wang[2], Hongqian Wang[2]
[1]College of Computer Science, Chongqing University, Chongqing, China
[2]The First Affiliated Hospital of Army Medical University, Chongqing, China
wangcl@cqu.edu.cn

***Abstract*—Camouflaged Object Detection (COD) aims to identify and segment objects that seamlessly blend into their surroundings. While the Segment Anything Model (SAM) excels in generic segmentation, its performance in COD is impeded by the significant domain gap between its general-purpose pretraining data and the ambiguous camouflaged targets. Existing SAM adaptations, categorized into *explicit prompting* and *architecture design*, attempt to bridge this gap but still suffer from two critical limitations: they often fail to maintain intra-object consistency, leading to fragmented segmentation, and struggle to delineate low-contrast boundaries. To address these challenges, we propose HPT-SAM2. Specifically, we design a Hierarchical Prototype Refinement (HPR) module that adopts a *global-to-local* strategy, utilizing coarse and fine-grained prototypes to consolidate scattered features into coherent entities. Additionally, a Texture-Semantic Modulation (TSM) module is developed to sharpen object contours by synergizing semantic feedback and texture priors via a gated mechanism. Experimental results show that HPT-SAM2 achieves a 4.2% improvement in weighted F-measure on COD10K and a 16.7% reduction in Mean Absolute Error on CAMO compared to the baseline SAM2-UNet.**

***Index Terms*—Camouflaged object detection, Segment Anything Model 2, Prototype learning, Feature modulation**

I. Introduction

Camouflaged Object Detection (COD) aims to segment objects that are visually blended into their surroundings. This task is inherently challenging due to the high visual similarity between the foreground and background. Advancing research in COD holds significant potential for a wide range of practical applications, including medical image analysis [1], industrial defect detection [2], and ecosystem monitoring [3].

Driven by the development of deep learning, researchers have proposed numerous methods [4]–[9] to tackle this challenge. While effective on specific datasets, traditional fully supervised methods often struggle to generalize to diverse or unseen scenes due to their heavy reliance on limited annotated data. To address this limitation, foundation models like the Segment Anything Model (SAM) [10] and SAM 2 [11] have emerged, revolutionizing segmentation with remarkable zero-shot capabilities. However, applying SAM directly to COD is impeded by a significant domain gap. SAM is pretrained on massive general-purpose datasets, whereas camouflaged targets exhibit ambiguous edges and low contrast, resulting in a severe data distribution mismatch.

Existing adaptations attempt to bridge this gap through two main paradigms: explicit prompt and architecture design approaches. Explicit prompting methods [12]–[15] mine discriminative cues to generate prompts but are often sensitive to noise, resulting in topological errors. Conversely, architecture design adaptations [16]–[19] typically inject lightweight adapters or redesign the decoder. Despite their progress, we argue that current solutions suffer from two critical limitations. First, they often **overlook intra-object feature consistency**. Consequently, the predicted masks tend to be fragmented, especially when the target is partially occluded, failing to preserve the semantic integrity of the object. Second, they struggle to handle the low foreground-background contrast, resulting in **blurred boundaries** and compromised spatial precision. To address these limitations, we propose **HPT-SAM2**, a novel framework designed to enhance semantic integrity and boundary precision. We introduce two key strategies to adapt SAM 2 for the COD task effectively. First, to mitigate semantic fragmentation, we propose the **Hierarchical Prototype Refinement (HPR)** module. Adopting a *global-to-local* strategy, HPR utilizes coarse prototypes to anchor global semantics and fine-grained prototypes to recover local details, consolidating scattered features into a coherent entity. Second, to resolve boundary ambiguity, we design the **Texture-Semantic Modulation (TSM)** module. TSM synergizes **semantic feedback** with **texture priors** via a gated mechanism, ensuring robust boundary recovery even in low-contrast scenarios. Our main contributions are summarized as follows:

- We propose the HPR module to consolidate intra-object consistency via a global-to-local prototype anchoring strategy, effectively suppressing segmentation fragmentation
- We design the TSM module to sharpen object boundaries by synergizing semantic feedback with low-level texture features, addressing the boundary ambiguity prevalent in existing methods.

*Corresponding author. This work was supported in part by Science and Technology Innovation Key RD Program of Chongqing(CSTB2025TIAD-STX0029).

- Experimental results show that HPT-SAM2 achieves a 4.2% improvement in weighted F-measure on COD10K and a 16.7% reduction in Mean Absolute Error on CAMO compared to the baseline SAM2-UNet.

II. Related work

A. Camouflaged Object Detection

Deep learning-based COD approaches typically fall into three paradigms: 1) Bio-inspired strategies [4], [5], [20] mimic the predatory "search and identify" mechanism, locating targets globally before progressively refining local details; 2) Multi-scale aggregation methods [21], [22] address extreme scale variations by integrating cross-level features to simultaneously capture semantic context and structural textures; and 3) Boundary and texture modeling approaches [6], [7], [23] utilize auxiliary edge or frequency supervision to amplify subtle discrepancies between ambiguous targets and homologous backgrounds.However, these specialized models are heavily dependent on limited annotated data, which significantly restricts their generalization to unseen scenes.

B. Segment Anything Model Adaptations

Although SAM [10] excels in generic segmentation, its application to COD is limited by the need for manual interaction and poor boundary sensitivity. Existing adaptations generally diverge into two paradigms: Explicit Prompting and Architecture Design.

1) Explicit Prompting. This paradigm aligns with SAM's interactive nature by automatically constructing task-specific guidance signals. Specifically, COMPrompter [12] adopts a multiprompt strategy, employing an Edge Gradient Extraction Module to synthesize boundary prompts that mutually refine box inputs for precise edge delineation. Similarly, DSAM [13] utilizes a Prompt-Deeper Module to filter RGB noise via depth cues, constructing robust depth-aware box prompts to drive the decoder. Furthermore, SAM-COD [15] extends this concept to the semantic level for RGB-D data. It employs dual-stream adapters to generate mixed prompt embeddings, effectively substituting external geometric prompts with implicit multimodal guidance signals.

2) Architecture Design. In contrast, this paradigm treats SAM as a foundational backbone, focusing on structural modifications to learn camouflaged representations directly without prompt engineering. **i) Feature Injection:** Methods like SAM-Adapter [16] insert lightweight adapters into the frozen encoder for parameter-efficient fine-tuning. MDSAM [17] enhances this by incorporating multi-scale aggregation to align object details at varying resolutions with pre-trained knowledge. **ii) Decoder Reconstruction:** Recent works redesign the decoding process for dense prediction using the SAM 2 backbone. SAM2-UNet [18] adopts a classic U-shaped architecture with skip connections to recover spatial details. Meanwhile, HFS-SAM2 [19] introduces a specialized frequency-aware decoder, integrating High-Frequency Supplementation modules to capture texture details and refine boundaries independent of the original prompt decoder.

Despite these advancements, most architecture design approaches still rely on simplistic aggregation mechanisms. They often overlook intra-object feature consistency, leading to semantic fragmentation, and fail to effectively sharpen the fuzzy boundaries inherent in low-contrast scenarios.

C. Overall Architecture

The architecture of HPT-SAM2 is illustrated in Fig. 1(a). Following the pipeline of SAM2-UNet [18], we utilize the SAM2 Hiera encoder equipped with lightweight Adapters and Receptive Field Blocks (RFB) to extract multi-scale features $\{F_i\}_{i=1}^{4}$. To rectify semantic fragmentation, the deepest feature F_4 is first processed by the Hierarchical Prototype Refinement (HPR) module to anchor global semantics. In the decoding phase, the Texture-Semantic Modulation (TSM) module is integrated at each upsampling stage, modulating features by leveraging the prediction mask from the previous stage and the original image as complementary guidance. Finally, the network generates multi-scale predictions $\{P_i\}_{i=1}^{4}$ via deep supervision.

III. Method

A. Hierarchical Prototype Refinement (HPR)

Motivation. Existing SAM-based methods often overlook intra-object feature consistency. Consequently, the predicted masks tend to be fragmented, especially when the target is partially occluded, failing to preserve the semantic integrity of the object. To rectify this topological defect, we draw inspiration from slot-attention [24], which excels at decomposing scenes into coherent entities via **slot-based clustering**. We hypothesize that the ambiguous camouflaged target can be effectively consolidated by treating segmentation as a **prototype-based clustering** process. Guided by this insight, **as illustrated in Fig. 1 (b)**, HPR adopts a *global-to-local* strategy. Instead of naive local aggregation, we explicitly utilize learnable prototypes as cluster centers to group similar semantic features—first anchoring global semantics with coarse prototypes and subsequently recovering local details with fine-grained ones. This mechanism ensures that scattered feature components are dynamically bound into a coherent entity.

1) Coarse-Grained Anchoring and Gating. To bridge disjointed semantic parts, we first establish a holistic estimate by clustering global semantics. Given the deepest feature input $F_4 \in \mathbb{R}^{C\times H\times W}$, we initialize $N_c = 2$ coarse prototypes $P_c \in \mathbb{R}^{N_c\times C}$ serving as global cluster centers. We aggregate these semantics via Slot Attention [24]:

$$S_c = SlotAttn(F_4, P_c). \tag{1}$$

Since slots are spatially abstract, we restore their spatial context by broadcasting S_c onto a 2D grid augmented with learnable spatial position embeddings P_e. These position-aware slots are fused via a 1×1 convolution to yield the global feature reconstruction $F_c \in \mathbb{R}^{C\times H\times W}$. Based on this global anchor, we explicitly suppress background noise to prepare for

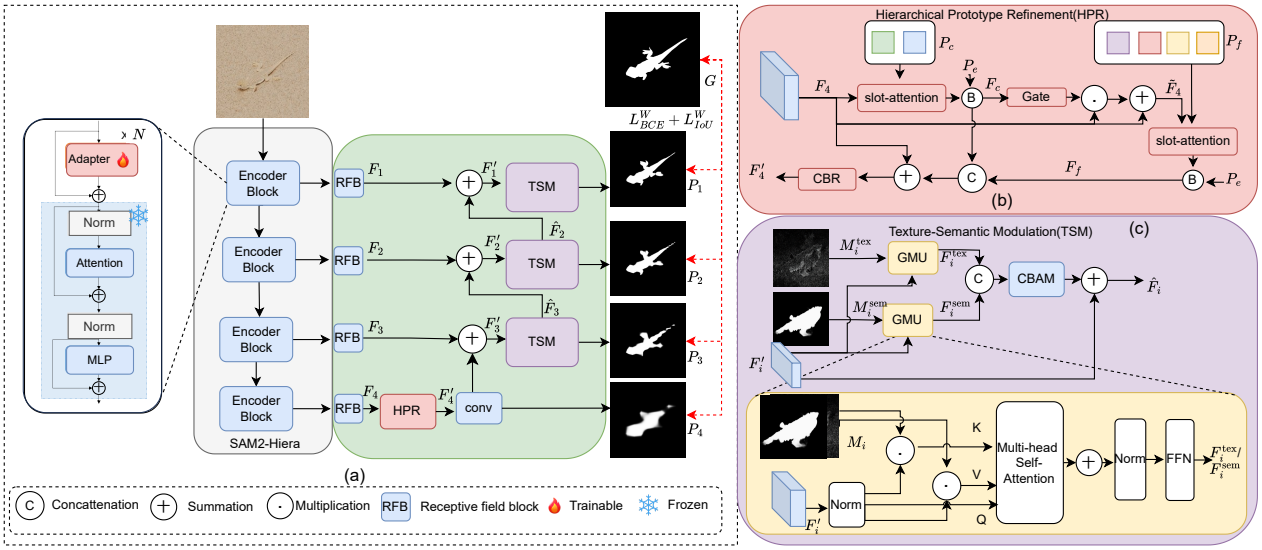

Fig. 1. The overall architecture of our proposed HPT-SAM2.

fine-grained refinement. A spatial attention gate $\mathcal{G}$ is derived from F_c to generate a purified feature $\tilde{F}_4$:

$$\mathcal{G} = \sigma(\mathcal{W}_g(F_c)), \quad \tilde{F}_4 = F_4 \odot (1 + \mathcal{G}). \tag{2}$$

Crucially, $\tilde{F}_4$ provides a focused search space with distractions suppressed, serving as the input for the next stage.

2) Fine-Grained Detail Refinement. Finally, to capture intricate boundaries and local structures, we deploy $N_f = 4$ fine prototypes P_f on the purified features $\tilde{F}_4$. We obtain fine slots S_f via:

$$S_f = SlotAttn(\tilde{F}_4, P_f). \tag{3}$$

Similar to the coarse stage, we reconstruct the fine-grained feature F_f by broadcasting the slots onto the position-augmented grid and fusing them. The final refined feature F'_4 integrates both the global anchor and local details via channel projection:

$$F'_4 = F_4 + \mathcal{W}_{out}(Concat(F_c, F_f)). \tag{4}$$

By explicitly enforcing consistency from coarse to fine granularity, HPR ensures that the object features in the deepest layer are preserved as a unified entity.

B. Texture-Semantic Modulation (TSM)

Motivation. In COD, targets often share intrinsic visual patterns with the background, leading to feature ambiguity during the decoding process. Relying solely on the upsampled features frequently results in blurred boundaries and lost spatial integrity. To address this challenge, **as illustrated in Fig. 1 (c)**, the TSM module is designed to sharpen decision boundaries by synergizing two complementary guidance signals: **semantic feedback** to anchor accurate object localization, and **texture priors** to refine fine-grained structural details. By explicitly forcing the decoding features to query these specific cues via a gated mechanism, we ensure both semantic correctness and boundary precision.

1) Dual-Source Guidance Construction. At each decoder stage i, we aim to enhance the input feature $F'_i \in \mathbb{R}^{C\times H\times W}$ by injecting priors from two distinct streams:

- **Semantic Feedback Stream:** To prevent false positives in background regions, we utilize the prediction P_{i+1} from the deeper stage. We upsample it to generate a high-level semantic mask $M_i^{sem} = \mathcal{U}(P_{i+1})$, which provides strong localization guidance.
- **Texture Prior Stream:** To sharpen object boundaries, we extract high-frequency structural cues from the original image I. We employ the Laplacian operator followed by resizing to obtain the texture prior $M_i^{tex} = \mathcal{R}(\nabla^2(I))$.

These two spatial maps, generically denoted as $M_i \in \{M_i^{sem}, M_i^{tex}\}$, serve as the attention gates for the subsequent modulation.

2) Gated Modulation Unit (GMU). To effectively transfer these spatial priors into the feature space, we design the Gated Modulation Unit (GMU), denoted as Ψ. Unlike standard cross-attention, GMU uses the spatial prior M_i to strictly constrain the attention field of the input feature F'_i. Specifically, we generate the Queries (Q) from the feature itself, while the Keys (K) and Values (V) are spatially gated by the prior M_i:

$$Q = \mathcal{W}_q(F'_i), \quad K = \mathcal{W}_k(F'_i) \odot M_i, \quad V = \mathcal{W}_v(F'_i) \odot M_i. \tag{5}$$

Here, $\odot$ denotes element-wise multiplication. By suppressing background noise in K and V, the attention mechanism forces the model to focus aggregation only on regions highlighted by the prior. The modulated feature is computed as:

$$\Psi(F'_i, M_i) = Softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{6}$$

We apply this operator to both streams to obtain the semantically-enhanced feature $F_i^{sem} = \Psi(F'_i, M_i^{sem})$ and the texture-sharpened feature $F_i^{tex} = \Psi(F'_i, M_i^{tex})$.

3) Adaptive Integration. Finally, since the importance of semantic positioning and texture details varies across different stages, we employ a split-attention fusion strategy. The two modulated features are concatenated and refined via a Convolutional Block Attention Module (CBAM) before being injected back into the residual stream:

$$\hat{F}_i = F_i' + CBAM(Concat(F_i^{sem}, F_i^{tex})). \tag{7}$$

The final modulated feature $\hat{F}_i$ is then projected via a 1×1 convolution to generate the prediction map P_i:

$$P_i = Conv_{1\times1}(\hat{F}_i). \tag{8}$$

This prediction facilitates deep supervision and serves as the semantic guidance for the next stage.

C. Loss Function

Our framework employs a structure-aware loss function that jointly optimizes pixel-level classification and global structure preservation. The training objective aggregates predictions from four decoding scales using a deep supervision strategy:

$$\mathcal{L}_{total} = \sum_{k=1}^{4} \lambda_k \left(\mathcal{L}_{bce}^{w}(P_k, G) + \mathcal{L}_{iou}^{w}(P_k, G)\right) \tag{9}$$

where P_k denotes the prediction map at the k-th scale and G represents the ground truth. $\mathcal{L}_{bce}^{w}$ and $\mathcal{L}_{iou}^{w}$ denote the weighted binary cross-entropy loss and weighted Intersection-over-Union loss, respectively, which utilize structure-aware weight maps to emphasize boundary pixels. The balancing coefficients are set as $\lambda = \{1.0, 0.5, 0.25, 0.125\}$ to prioritize the highest-resolution output while enforcing hierarchical consistency.

IV. Experiments

A. Experimental Setup

1) Datasets: We evaluate our method on four well-established COD benchmarks: CAMO [25], COD10K [4], CHAMELEON [26], and NC4K [27]. Following previous studies [4], [6], [7], we use 1,000 images from the CAMO dataset and 3,040 images from the COD10K dataset for training, with the remaining images used for testing.

2) Evaluation Metrics: Following [6], [8], [23], we use four widely adopted metrics for evaluation: Structure-measure (S_α) [28], Mean Absolute Error (M), Weighted F-measure (F_β^w) [29], and Mean E-measure (E_ϕ) [30].

3) Implementation Details: We implement our model using PyTorch and conduct experiments on a single NVIDIA RTX 4090 GPU (24 GB). We use the SAM2-Hiera-L as the encoder. Input images are resized to 352×352 pixels to maintain consistency across the dataset. For optimization, we employ the AdamW optimizer with a weight decay of 1×10^{-3}. The batch size is set to 8. The learning rate is initialized at 2×10^{-4} and follows a cosine decay strategy for a total of 50 epochs.

B. Comparison with State-of-the-art Methods

To demonstrate the effectiveness of HPT-SAM2, we conduct a comprehensive comparison with 16 state-of-the-art methods. These methods are categorized into three groups: 1) Specialized COD models [5]–[9], [20]–[23]; 2) Explicit prompting SAM adaptations [12]–[15]; and 3) Architecture design SAM adaptations [16]–[19]. Table I summarizes the quantitative results.

1) Comparison with specialized COD models. HPT-SAM2 consistently outperforms specialized models across all benchmarks. Notably, on the challenging CAMO dataset, it sets a new record with an F_β^w of **0.873** and reduces the Mean Absolute Error ($\mathcal{M}$) by approximately **16.7%** compared to the Camouflageator [20].

2) Comparison with explicit prompting SAM adaptations. While methods like COMPrompter [12] exhibit competitive performance on COD10K and NC4K, HPT-SAM2 demonstrates distinct superiority on CAMO and CHAMELEON datasets.These results validate that effective architecture design serves as a powerful alternative to explicit prompting, capable of achieving comparable or even superior adaptation efficacy.

3) Comparison with architecture design SAM adaptations. HPT-SAM2 comprehensively surpasses the baseline SAM2-UNet [18] across all four benchmarks. Notably, we achieve a 4.2% improvement in F_β^w on COD10K and a 16.7% reduction in error rate ($\mathcal{M}$) on CAMO. These consistent gains validate that our HPR and TSM modules effectively resolve the semantic fragmentation and boundary ambiguity inherent in the baseline.

C. Ablation Study

To verify the contribution of HPT-SAM2, we conduct ablation experiments on four benchmarks using SAM2-UNet [18] as the baseline. The quantitative results are reported in Table II.

1) Effectiveness of HPR. As shown in the second row of Table II, HPR significantly mitigates semantic fragmentation. On the challenging CAMO dataset, it boosts S_α from 0.884 to 0.899 and reduces the error rate ($\mathcal{M}$) by 11.9% compared to the baseline. This confirms that our global-to-local strategy effectively consolidates scattered features into coherent entities.

2) Effectiveness of TSM. Referring to the third row of Table II, TSM excels at resolving boundary blurring via edge injection. On COD10K, it improves F_β^w from 0.789 to 0.803. This validates that injecting Laplacian priors helps distinguish subtle texture breaks, sharpening details often smoothed by the baseline.

3) Synergistic Efficacy. Integrating both modules yields the best performance (fourth row of Table II), boosting F_β^w on COD10K by 4.2%. This significant gain validates our design philosophy: HPR ensures semantic integrity, providing a reliable anchor that allows TSM to focus on refining precise boundaries without distraction.

4) Hyperparameter Analysis. We further investigate the prototype numbers (N_c, N_f) as shown in Table III. Results

TABLE I

QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON FOUR BENCHMARK DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Method	Pub	CAMO				COD10K				CHAMELEON				NC4K			
		$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
Specialized COD Models																	
BGNet [6]	IJCAI 22	0.812	0.870	0.749	0.073	0.831	0.901	0.722	0.033	-	-	-	-	0.851	0.907	0.788	0.044
ZoomNet [5]	CVPR 22	0.820	0.892	0.752	0.066	0.838	0.911	0.729	0.029	0.902	0.958	0.845	0.023	0.853	0.912	0.784	0.043
FEDER [7]	CVPR 23	0.836	0.897	0.807	0.066	0.844	0.911	0.748	0.029	0.903	0.947	0.856	0.026	0.862	0.913	0.824	0.042
CamoFormer-R [21]	TPAMI 24	0.817	0.884	0.756	0.066	0.838	0.898	0.730	0.029	-	-	-	-	0.857	0.915	0.793	0.041
ZoomNeXT-R [22]	TPAMI 24	0.833	0.891	0.774	0.065	0.861	0.925	0.768	0.026	0.908	0.963	0.858	0.021	0.874	0.928	0.816	0.037
Camouflageator [20]	ICLR 24	0.871	0.931	0.861	0.042	0.862	0.934	0.788	0.023	0.908	0.961	0.867	0.022	0.883	0.937	0.827	0.042
CamoDifussion [8]	TPAMI 25	0.878	0.940	0.853	0.042	0.881	0.944	0.814	0.020	-	-	-	-	0.893	0.942	0.859	0.029
RUN [9]	ICML 25	0.877	0.934	0.861	0.045	0.878	0.941	0.810	0.021	0.877	0.934	0.861	0.045	0.892	0.940	0.868	0.030
ESC-Net [23]	ICCV 25	0.871	0.934	0.843	0.044	0.873	0.939	0.804	0.021	-	-	-	-	0.892	0.941	0.859	**0.028**
Explicit Prompting SAM Adaptations																	
COMPrompter [12]	SCIS 24	0.882	0.942	0.858	0.044	0.889	0.949	0.821	0.023	0.906	0.955	0.857	0.026	**0.907**	0.955	**0.876**	0.030
DSAM [13]	ACMMM 24	0.832	0.913	0.794	0.061	0.846	0.931	0.760	0.033	-	-	-	-	0.871	0.932	0.826	0.040
COD-SAM [14]	PR 25	0.870	0.906	0.796	0.055	**0.899**	0.941	0.832	0.021	**0.924**	0.956	0.880	0.021	-	-	-	-
SAM-DSA [15]	ICCV 25	0.866	**0.948**	0.839	0.047	0.881	**0.969**	0.817	0.023	0.888	**0.965**	0.841	0.031	0.889	**0.963**	0.857	0.032
Architecture Design SAM Adaptations																	
SAM-Adapter [16]	ICCVW 2023	0.847	0.873	0.765	0.070	0.883	0.918	0.801	0.025	0.896	0.919	0.824	0.033	-	-	-	-
MDSAM [17]	ACMMM 24	0.852	0.903	0.834	0.053	0.862	0.921	0.803	0.025	-	-	-	-	0.875	0.921	0.850	0.037
SAM2-Unet [18]	arXiv 24	0.884	0.932	0.861	0.042	0.880	0.936	0.789	0.021	0.914	0.961	0.863	0.022	0.901	0.941	0.863	0.029
HFS-SAM2 [19]	SPL 25	0.883	0.922	0.839	0.045	0.879	0.913	0.789	0.023	0.909	0.943	0.849	0.023	0.900	0.925	0.849	0.030
HPT-SAM2 (Ours)	-	**0.900**	0.945	**0.873**	**0.035**	0.888	0.944	0.822	**0.019**	**0.924**	**0.965**	**0.887**	**0.019**	0.904	0.944	0.867	**0.028**

TABLE II

ABLATION STUDY OF THE PROPOSED COMPONENTS ON FOUR BENCHMARK DATASETS. "HPR" DENOTES THE HIERARCHICAL PROTOTYPE REFINEMENT MODULE, AND "TSM" DENOTES THE TEXTURE-SEMANTIC MODULATION MODULE. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**.

Components		CAMO				COD10K				CHAMELEON				NC4K			
HPR	TSM	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
		0.884	0.932	0.861	0.042	0.880	0.936	0.789	0.021	0.914	0.961	0.863	0.022	0.901	0.941	0.863	0.029
✓		0.899	0.940	0.861	0.037	0.882	0.935	0.802	0.021	0.919	0.960	0.871	0.020	0.905	0.943	0.861	0.028
	✓	0.888	0.932	0.860	0.041	0.880	0.936	0.803	0.021	0.908	0.952	0.863	0.022	0.900	0.938	0.858	0.029
✓	✓	**0.900**	**0.945**	**0.873**	**0.035**	**0.888**	**0.944**	**0.822**	**0.019**	**0.924**	**0.965**	**0.887**	**0.019**	**0.904**	**0.944**	**0.867**	**0.028**

TABLE III

ABLATION STUDY ON THE HYPERPARAMETERS OF THE HPR MODULE (N_c AND N_f) ON CAMO AND COD10K DATASETS. N_c AND N_f REPRESENT THE NUMBER OF COARSE AND FINE PROTOTYPES, RESPECTIVELY.

Settings		CAMO				COD10K			
N_c	N_f	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{S}_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
2	4	**0.900**	**0.945**	**0.873**	**0.035**	**0.888**	**0.944**	**0.822**	**0.019**
2	8	0.898	0.942	0.866	0.035	0.885	0.941	0.814	0.020
4	8	0.900	0.942	0.866	0.035	0.886	0.937	0.810	0.020

indicate that $N_c = 2$ and $N_f = 4$ yield optimal performance. Increasing fine prototypes ($N_f = 8$) drops F_β^w on COD10K to 0.814, likely due to over-segmentation of texture noise. Similarly, increasing coarse prototypes ($N_c = 4$) further degrades F_β^w to 0.810, suggesting that additional coarse anchors introduce semantic ambiguity.

D. Qualitative Visualization

1) Visual Comparison with SOTA Methods. Figure 2 compares HPT-SAM2 with state-of-the-art methods. In complex scenarios like heavy occlusion (Row 4), **COMPrompter** tends to generate segmentation results containing internal holes, while the baseline **SAM2-UNet** fails to completely segment the camouflaged target, missing significant regions. In contrast, HPT-SAM2 maintains superior object integrity, validating the global anchoring capability of our HPR module. Furthermore, for targets with thin structures (Row 5) or ambiguous boundaries (Row 3), our method delineates significantly sharper details compared to the over-smoothed predictions of SAM2-UNet.

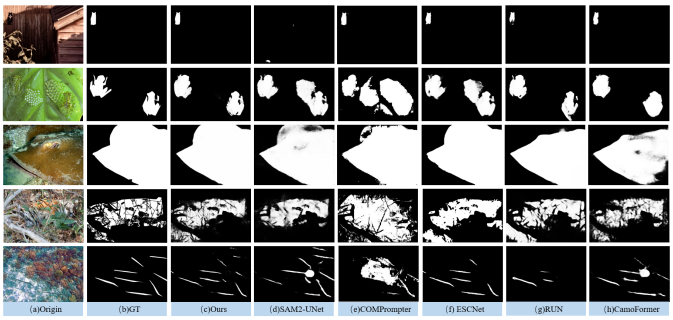

Fig. 2. Visual example comparisons between our HPD-SAM2 and some previous SOTA methods.

2) Visualization of Hierarchical Prototypes. Figure 3 illustrates the attention maps of the HPR module. The Coarse

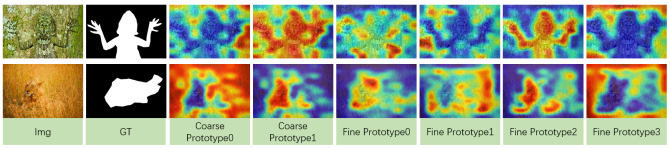

Fig. 3. Visualization of coarse and fine prototypes

Prototypes act as global anchors, locating the main object body while suppressing background noise. The Fine Prototypes then refine local details to ensure completeness. As shown in the second row, while coarse prototypes localize the general region, Fine Prototype 2 specifically enhances the feature consistency between the lion's head and its occluded body, effectively bridging the semantic gap caused by occlusion.

V. Conclusion

In this paper, we present HPT-SAM2 to bridge the significant domain gap between the general-purpose SAM 2 and the intricate challenges of Camouflaged Object Detection (COD). To mitigate semantic fragmentation, the proposed Hierarchical Prototype Refinement (HPR) module employs a global-to-local strategy, leveraging coarse and fine-grained prototypes to consolidate scattered features into coherent entities. Additionally, our Texture-Semantic Modulation (TSM) module resolves boundary ambiguity by dynamically synergizing semantic feedback with texture priors via a gated mechanism. Extensive evaluations on standard benchmarks demonstrate that HPT-SAM2 establishes a robust, state-of-the-art framework for high-mimicry scenarios.

References

[1] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Pranet: Parallel reverse attention network for polyp segmentation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2020, pp. 263–273.

[2] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE transactions on industrial electronics*, vol. 55, no. 1, pp. 348–363, 2008.

[3] N. Price, S. Green, J. Troscianko, T. Tregenza, and M. Stevens, "Background matching and disruptive coloration as habitat-specific strategies for camouflage," *Scientific reports*, vol. 9, no. 1, p. 7840, 2019.

[4] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2777–2787.

[5] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022, pp. 2160–2170.

[6] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," *arXiv preprint arXiv:2207.00794*, 2022.

[7] C. He, K. Li, Y. Zhang, L. Tang, Y. Zhang, Z. Guo, and X. Li, "Camouflaged object detection with feature decomposition and edge reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 046–22 055.

[8] Z. Chen, K. Sun, and X. Lin, "Camodiffusion: Camouflaged object detection via conditional diffusion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 2, 2024, pp. 1272–1280.

[9] X. Zhang, B. Yin, Z. Lin, Q. Hou, D.-P. Fan, and M.-M. Cheng, "Referring camouflaged object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[11] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[12] X. Zhang, Z. Yu, L. Zhao, D.-P. Fan, and G. Xiao, "Comprompter: reconceptualized segment anything model with multiprompt network for camouflaged object detection," *Science China Information Sciences*, vol. 68, no. 1, p. 112104, 2025.

[13] Z. Yu, X. Zhang, L. Zhao, Y. Bin, and G. Xiao, "Exploring deeper! segment anything model with depth perception for camouflaged object detection," in *Proceedings of the 32nd ACM international conference on multimedia*, 2024, pp. 4322–4330.

[14] D. Gao, Y. Zhou, H. Yan, C. Chen, and X. Hu, "Cod-sam: Camouflage object detection using sam," *Pattern Recognition*, p. 111826, 2025.

[15] J. Liu, L. Kong, and G. Chen, "Improving sam for camouflaged object detection via dual stream adapters," *arXiv preprint arXiv:2503.06042*, 2025.

[16] T. Chen, L. Zhu, C. Deng, R. Cao, Y. Wang, S. Zhang, Z. Li, L. Sun, Y. Zang, and P. Mao, "Sam-adapter: Adapting segment anything in underperformed scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3367–3375.

[17] S. Gao, P. Zhang, T. Yan, and H. Lu, "Multi-scale and detail-enhanced segment anything model for salient object detection," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 9894–9903.

[18] X. Xiong, Z. Wu, S. Tan, W. Li, F. Tang, Y. Chen, S. Li, J. Ma, and G. Li, "Sam2-unet: Segment anything 2 makes strong encoder for natural and medical image segmentation," *arXiv preprint arXiv:2408.08870*, 2024.

[19] Z. Wu, X. Xiong, G. Gao, H. Li, and H. Chen, "Hfs-sam2: Segment anything model 2 with high-frequency feature supplementation for camouflaged object detection," *IEEE Signal Processing Letters*, 2025.

[20] C. He, K. Li, Y. Zhang, Y. Zhang, Z. Guo, X. Li, M. Danelljan, and F. Yu, "Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects," *arXiv preprint arXiv:2308.03166*, 2023.

[21] B. Yin, X. Zhang, D.-P. Fan, S. Jiao, M.-M. Cheng, L. Van Gool, and Q. Hou, "Camoformer: Masked separable attention for camouflaged object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[22] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoomnext: A unified collaborative pyramid network for camouflaged object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 12, pp. 9205–9220, 2024.

[23] S. Ye, X. Chen, Y. Zhang, X. Lin, and L. Cao, "Escnet: Edge-semantic collaborative network for camouflaged object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 20 053–20 063.

[24] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," *Advances in neural information processing systems*, vol. 33, pp. 11 525–11 538, 2020.

[25] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Computer vision and image understanding*, vol. 184, pp. 45–56, 2019.

[26] S. Przemysław, A. Hassan, B. Jakub, D. Tomasz, K. Adam, and P. Kozieł, "Animal camouflage analysis: Chameleon database," *Unpublished manuscript*, 2018.

[27] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 591–11 601.

[28] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4548–4557.

[29] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 248–255.

[30] D.-P. Fan, G.-P. Ji, X. Qin, and M.-M. Cheng, "Cognitive vision inspired object segmentation metric and loss function," *Scientia Sinica Informationis*, vol. 6, no. 6, p. 5, 2021.