Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MC-SAM: A Stability-Constrained Coupled Adaptation Framework for SAM in Camouflaged Scene Segmentation

Zihua Liu
South China Normal University
Yonghao Wu
2033902013@m.scnu.edu.cn
South China Normal University
Jiajie Murong
South China Normal University
Chang Liu
Guangdong Ocean University

Research Article

Keywords: Camouflaged object segmentation, Segment Anything Model, fusion stability, doubly stochastic mixing, multimodal adaptation

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11209467/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

MC-SAM: A Stability-Constrained Coupled Adaptation Framework for SAM in Camouflaged Scene Segmentation

Zihua Liu[1], Yonghao Wu[1,*], Jiajie Murong[1], Chang Liu[2]

[1]School of Electronic Science and Engineering (School of Microelectronics),
South China Normal University, Foshan 528000, China
[2]St. Petersburg School of Shipbuilding and Ocean Technology,
Guangdong Ocean University, Zhanjiang 524088, China

[*]Corresponding author: 2023025071@m.scnu.edu.cn

Abstract

Camouflaged object segmentation depends on weak foreground cues, yet SAM adaptations chiefly enrich those cues without controlling how heterogeneous streams mix. MC-SAM addresses this fusion problem with a shared doubly stochastic prior at two interfaces: CSPG forms sparse prompts from visual and textual cues, while MCA adapts intermediate SAM features. Sinkhorn normalization constrains both output mixtures and aggregate input contributions; the exact mixer has an operator-level non-expansiveness bound. Factorized ablations show gains before HyperCond conditioning or optional RankDice-RMA mask selection. In matched COD10K comparisons, Sinkhorn fusion raises weighted F-measure from .842 to .858, reduces feature distance distortion from .25 to .07, and lowers threshold sensitivity from .026 to .012. MC-SAM trains 5.151M parameters with a frozen ViT-H image encoder. Together, results across four COD benchmarks and matched fusion comparisons support constrained fusion as a practical means of preserving weak foreground cues under camouflage. The theoretical guarantee concerns the exact mixing operator; full-network stability is assessed empirically rather than asserted as a guarantee.

Keywords: Camouflaged object segmentation; Segment Anything Model; fusion stability; doubly stochastic mixing; multimodal adaptation.

1 Introduction

Camouflaged object detection (COD), considered here as pixel-wise segmentation, depends on weak distinctions between foreground objects and visually similar backgrounds. SAM-based methods address this difficulty through domain adaptation and richer visual or multimodal prompts [5, 21, 44]. These advances improve the evidence available to the model. Their effectiveness also depends on whether fusion preserves that evidence when heterogeneous streams interact.

Under camouflage ambiguity, a small change in feature relationships can erase a useful foreground cue or reinforce a misleading background response. Unconstrained mixing can amplify dominant branches, distort local geometry, and increase sensitivity to the final decision threshold. Figure 1 illustrates these changes in feature distances and norms. This motivates examining fusion stability as an explicit design objective alongside cue construction.

Our hypothesis is that regulating cross-stream mixing at both SAM adaptation interfaces improves how the model uses its available cues. Row normalization controls each output mixture, while column normalization constrains the total contribution of each input stream. Sinkhorn normalization combines these requirements through an approximate doubly stochastic matrix [6].

The exact mixing operator is non-expansive; the benefits of its finite-iteration, input-dependent implementation must be assessed in the complete segmentation system.

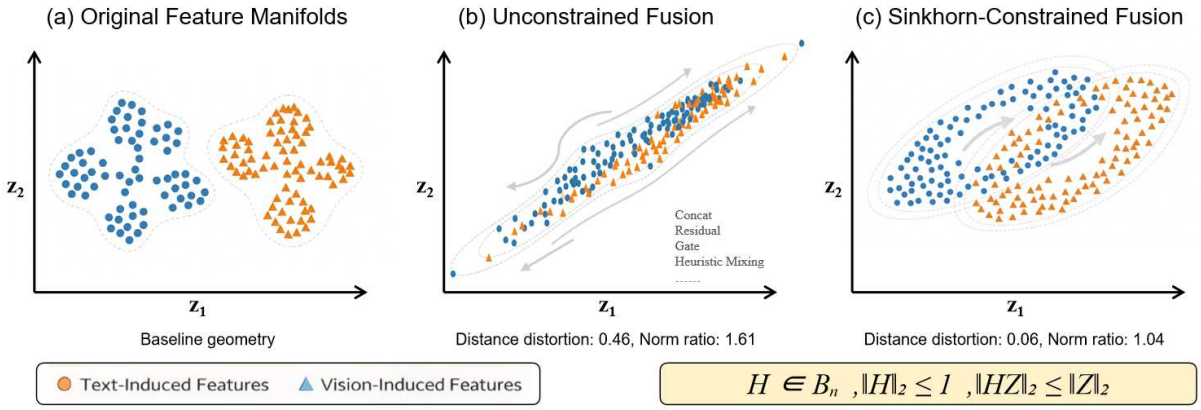


Fig. 1 Representative feature diagnostics before and after cross-stream fusion. The comparison illustrates changes in feature distances and norms under unconstrained and Sinkhorn-constrained mixing. Distance distortion measures relative changes in normalized neighborhood distances; norm ratio compares post-fusion and pre-fusion feature norms. These diagnostics characterize the displayed representations, while the theoretical bound in Section 3.3 concerns the mixing operator.

We propose MC-SAM, a stability-constrained coupled adaptation framework built around one shared structural prior at two fusion interfaces. CSPG constrains the mixing of textual and visual cues into sparse prompts; MCA constrains residual-stream mixing within SAM. Their coupling is structural: both impose the same row-and-column normalization principle, while retaining separate learned mixing matrices. This design makes fusion itself the intervention tested by the ablations and matched operator comparisons.

HyperCond and RankDice-RMA support this core rather than define additional fusion mechanisms (Fig. 2). HyperCond injects threshold conditioning into the dense prompt; the boundary weight λ_b controls training supervision only. RankDice-RMA optionally refines the decoder output during inference and adds no training loss. MC-SAM retains the frozen 637M SAM ViT-H backbone and trains 5.151M adaptation parameters. The experiments follow Q1/Q2/Q3: overall competitiveness, attribution to the fusion core, and the incremental roles of the auxiliary components. The factorized ablations and matched fusion comparisons provide the main attribution evidence.

Our contributions are threefold:

1. We formulate fusion stability as an explicit adaptation objective for SAM-based COD, focusing on whether heterogeneous mixing preserves weak foreground evidence.
2. We introduce coupled prompt-space and feature-space fusion through CSPG and MCA, with a shared doubly stochastic prior regulating both output mixtures and input contributions.
3. We test the fusion design with factorized ablations and matched operator comparisons. These experiments connect its segmentation gains to reduced distortion and threshold sensitivity, while separately evaluating auxiliary conditioning and output selection.

2 Related Work

Research on COD has progressed from handcrafted appearance cues [38, 41] to learned representations for target localization and boundary recovery [55]. We review four relevant directions,

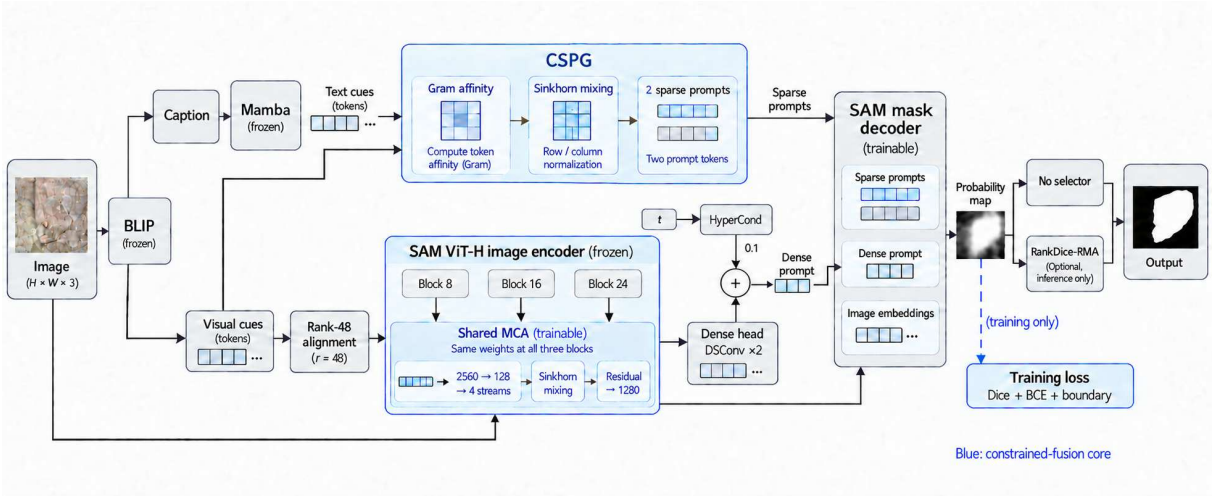


Fig. 2 Architecture of MC-SAM. Frozen BLIP and Mamba extract visual and textual cues. CSPG uses Gram-affinity Sinkhorn mixing to produce two sparse prompts. Rank-48 alignment supplies visual features to one trainable MCA with fully shared weights at blocks 8, 16, and 24 of the frozen SAM ViT-H encoder. MCA compresses concatenated features from 2560 to 128 dimensions, mixes four streams, and projects the residual to 1280 dimensions. A two-stage depthwise-separable dense head and the scaled HyperCond contribution form the dense prompt. The trainable mask decoder receives sparse prompts, the dense prompt, and image embeddings. The no-selector path exports the continuous probability map used for the main benchmark; optional, parameter-free RankDice-RMA selects a binary mask from that map. The dashed arrow denotes training supervision. Input, probability, and output thumbnails are schematic illustrations rather than quantitative results.

focusing on cue construction and the fusion of those cues within segmentation models.

2.1 CNN-based architectures for COD

Large annotated datasets have supported the development of deep COD models [9, 23]. Bio-inspired approaches, including SINet, PFNet, and PreyNet, organize prediction around search and identification processes [9, 37, 61]. Multi-scale methods combine features to recover target structure and boundaries [4, 13, 40, 52, 64]. Joint-learning approaches exploit related objectives, including ranking and uncertainty estimation [8, 23, 24, 36]. Other CNN designs improve aggregation and contextual reasoning [3, 10, 30, 39, 53, 54].

These methods establish the importance of combining target-level semantics with local structural information. MC-SAM examines the mixing operation used to incorporate such information into a pretrained segmentation model.

2.2 Transformer-based and hybrid architectures for COD

Attention mechanisms support modeling of long-range visual context [48]. UGTR represents uncertainty in camouflaged regions [56], while FSPNet aggregates features through progressive channel compression [18]. HitNet uses iterative high-resolution feedback [16], and CamoFormer separates foreground and background information through masked attention [57]. OSFormer extends attention-based modeling to camouflaged instance segmentation [42]. Additional models combine multi-scale, complementary, or decoupled representations [19, 20, 34, 39, 63, 66].

These architectures provide strong task-specific representations. Our focus is the integration of external cues into SAM while retaining its original image-encoder parameters. Specifically, we investigate how cross-stream constraints affect the fusion used during adaptation.

2.3 Segment Anything Model and its adaptations

SAM supports segmentation through image features and prompts [21]. SAM-Adapter introduces domain-specific adaptation modules [5], and ASAM investigates adversarial adaptation [25]. GenSAM and ProMaC explore automated prompting using foundation-model cues [14, 15].

MM-SAM combines visual and textual information for camouflage segmentation [44]. Related work investigates dual-stream adapters, mask guidance, and text-driven semi-supervised learning [22, 33, 50]. MC-SAM retains the multimodal cue extraction, adapter-based adaptation, and dense-prompt pathway of this line of work; their presence alone is not our claimed novelty. The distinction lies in how heterogeneous streams are mixed: CSPG applies Sinkhorn-constrained row-and-column mixing when constructing sparse prompts, while a single weight-shared bottleneck MCA applies the same structural prior to spatial feature adaptation at three encoder blocks. HyperCond and the optional output selector have supporting roles. The factorized ablations and matched fusion-operator comparisons test the contribution of this constrained-fusion design rather than treating the use of multiple cues or adapters as new.

2.4 Vision-language prompting for segmentation

Vision-language models provide semantic cues that can guide dense prediction. CLIP learns representations from paired images and text [43]; subsequent work studies localization and explainability [27, 28]. BLIP provides image captioning and visual representations through language-image pretraining [26]. Prompt-based segmentation methods use these capabilities to reduce dependence on manual spatial input [14, 34].

MC-SAM uses BLIP and Mamba [12] as upstream cue extractors. Its methodological focus is how those cues enter sparse prompts and intermediate visual features. The proposed constraint regulates cross-stream contributions at both locations, while the cue extractors provide the input representations.

3 Methodology

3.1 Overview of Architecture

MC-SAM centers on a shared constrained-fusion mechanism in two pathways (Fig. 2). BLIP [26] generates a caption and visual embeddings, and Mamba [12] encodes the caption. CSPG fuses global textual and visual cues into sparse prompts. A rank-48 factorized projection aligns the BLIP visual features for feature adaptation. The resulting representation is reused by a single bottleneck MCA that is invoked at three selected SAM encoder blocks with all parameters shared across the insertion sites.

Both pathways regulate cross-stream contributions using Sinkhorn normalization. CSPG supplies semantic guidance to the decoder, while MCA adapts the spatial representation in a 128-dimensional latent space before projecting the residual back to the 1280-dimensional SAM width. A two-stage depthwise-separable dense head and a lightweight spatial HyperCond projection provide the conditional dense prompt. The original SAM ViT-H image encoder remains frozen, whereas the mask decoder and compact adaptation modules are trained, giving 5.151M trainable parameters. The fusion mechanism is developed first; conditioning, training, and output selection are described afterward.

3.2 The Shared Fusion Mechanism in Prompt Space

CSPG implements the prompt-space component of the framework (Fig. 3). Textual and visual cues describe the same scene through different representations. We project them into a common prompt dimension and constrain their mixing across two streams. Row and column normalization regulate output allocation and aggregate input contribution, respectively.

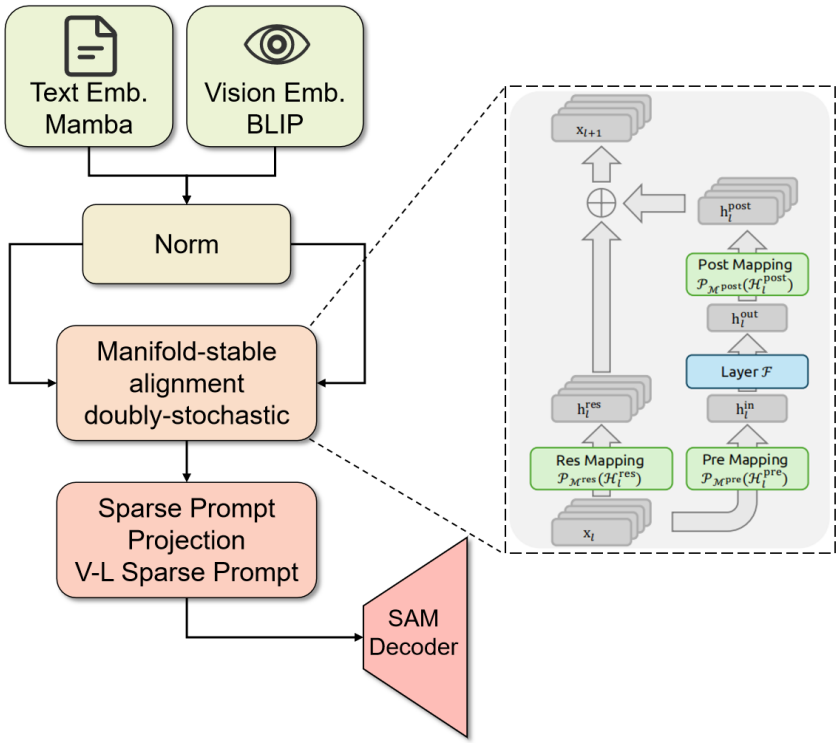


Fig. 3 CSPG with Gram-affinity mixing. Row-normalized embeddings define $\mathbf{A} = \mathbf{U}\mathbf{U}^T$. Sinkhorn normalization produces the pre-mixing matrix. Learned column-normalized post-mapping combines the pre-mixed streams into two aligned prompts.

Let $\mathbf{e}^{\text{text}} \in \mathbb{R}^{d_t}$ and $\mathbf{e}^{\text{vis}} \in \mathbb{R}^{d_v}$ denote textual and visual embeddings. Learned projections align them to dimension d, forming two prompt streams $\mathbf{P} \in \mathbb{R}^{2\times d}$. Row normalization gives $U_i = P_i/\|P_i\|_2$ for nonzero rows. We construct the affinity and positive kernel as

$$\mathbf{A} = \mathbf{U}\mathbf{U}^T \in \mathbb{R}^{2\times 2}, \qquad \mathbf{K} = \exp(\mathbf{A}/\epsilon), \quad \epsilon > 0, \tag{1}$$

where the exponential is element-wise and ϵ controls affinity sharpness. This construction derives stream-level affinity directly from the aligned embeddings.

Sinkhorn–Knopp normalization [6] alternates row and column scaling. For positive scaling vectors u and v,

$$u^{(j+1)} = \mathbf{1} \oslash (\mathbf{K}v^{(j)}), \qquad v^{(j+1)} = \mathbf{1} \oslash (\mathbf{K}^T u^{(j+1)}). \tag{2}$$

The resulting matrix is $\mathbf{H}_{\text{pre}} = \text{diag}(u)\mathbf{K}\,\text{diag}(v)$. Alternating normalization approaches the doubly stochastic constraint, which regulates both output allocation and aggregate input contribution. The pre-mixed prompts are

$$\mathbf{P}_{\text{mix}} = \mathbf{H}_{\text{pre}}\mathbf{P} \in \mathbb{R}^{2\times d}. \tag{3}$$

Post-mapping uses a learned stream-score function $g_\theta : \mathbb{R}^{2\times d} \to \mathbb{R}^{2\times 2}$. Let $\mathbf{B} = g_\theta(\mathbf{P}_{\text{mix}})$. Column normalization and the aligned output are defined explicitly as

$$(H_{\text{post}})_{ij} = \frac{\exp(B_{ij})}{\sum_{k=1}^{2}\exp(B_{kj})}, \qquad \mathbf{P}_{\text{aligned}} = \mathbf{H}_{\text{post}}^T\mathbf{P}_{\text{mix}}. \tag{4}$$

Both mixing matrices therefore act on the stream axis, and the output retains two d-dimensional prompt tokens. Each output token is a convex combination of the pre-mixed tokens. The probability-simplex constraint applies to its mixing weights; the prompt coordinates themselves need not form a probability vector. The operator bound below concerns $\mathbf{H}_{\text{pre}}$, not an unrestricted composition with post-mapping.

Row-softmax alone regulates each output allocation, while column normalization also regulates the total input contribution. This distinction motivates the shared constrained-fusion principle in CSPG and MCA.

3.3 The Shared Fusion Mechanism in Feature Space

MCA applies the same mixing principle to intermediate SAM features (Fig. 4). Sparse prompts influence decoder guidance, whereas feature adaptation acts on the spatial representation used for localization. To reduce the cost of this feature-space path, all stream construction and constrained mixing are performed inside a narrow latent space, and the same MCA parameters are reused at every insertion site.

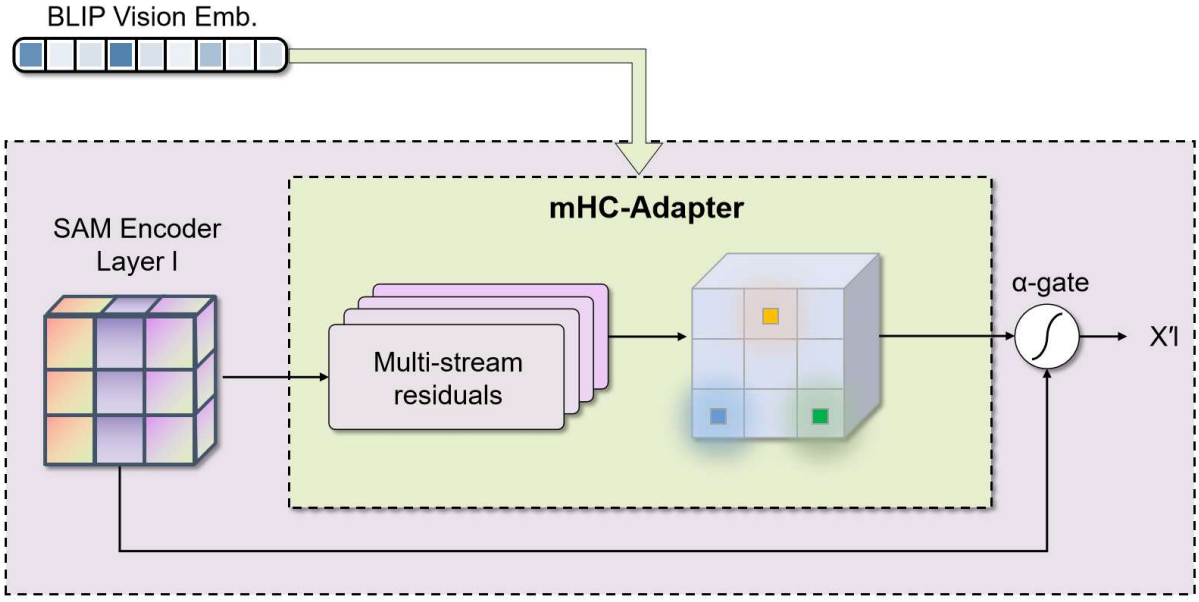


Fig. 4 Conceptual MCA residual path. In the implemented bottleneck variant, aligned BLIP and SAM features are concatenated and projected to $d = 128$ before four-stream generation and token-wise constrained mixing. Softmax-gated aggregation is projected back to the 1280-dimensional SAM width, and residual scaling preserves the original SAM feature through the skip connection. One set of MCA weights is shared across all three insertion sites.

With the ViT-H backbone, the shared MCA is inserted after blocks $l \in \{8, 16, 24\}$. Let $\mathbf{x}_p^{(l)} \in \mathbb{R}^{1280}$ denote the SAM feature at spatial token p, and let $\mathbf{v}_p^{(l)} \in \mathbb{R}^{1280}$ denote the aligned BLIP feature. The alignment map is factorized with rank $r = 48$, and its output is reused across the three MCA calls. MCA first forms a bottleneck representation

$$\mathbf{z}_p^{(l)} = \sigma\Big(\mathbf{W}_{\text{down}}[\mathbf{x}_p^{(l)}; \mathbf{v}_p^{(l)}] + \mathbf{b}_{\text{down}}\Big) \in \mathbb{R}^d, \qquad \mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times 2560}, \quad d = 128, \tag{5}$$

where σ is GELU. Four branch functions ϕ_s operate only in this latent space, giving $R_{psa} = [\phi_s(\mathbf{z}_p^{(l)})]_a$ for $s \in \{1, \ldots, 4\}$ and $a \in \{1, \ldots, d\}$. A lightweight mixing network predicts a token-wise cost matrix $\mathbf{C}_p \in \mathbb{R}^{4\times 4}$, giving

$$\mathbf{H}_p = \mathrm{SK}_{20}(\exp(-\mathbf{C}_p)), \tag{6}$$

where SK_{20} denotes twenty alternating row and column normalization steps. The mixed bottleneck residual and the SAM-width output are

$$\widetilde{\mathbf{z}}_p^{(l)} = \sum_{s=1}^{4} q_s \sum_{j=1}^{4} H_{psj}\mathbf{R}_{pj}, \qquad q = \mathrm{softmax}(\alpha), \qquad \mathbf{x}_p^{(l),\text{out}} = \mathbf{x}_p^{(l)} + 0.1\mathbf{W}_{\text{up}}\widetilde{\mathbf{z}}_p^{(l)}, \tag{7}$$

where $\mathbf{W}_{\text{up}} \in \mathbb{R}^{1280\times d}$. The down- and up-projection weights contribute approximately $2560 \times 128 + 128 \times 1280 = 0.492$M parameters, excluding biases and the lightweight branch and mixer layers. Crucially, the parameters $\{\mathbf{W}_{\text{down}}, \mathbf{W}_{\text{up}}, \phi_s, \alpha, \text{mixer}\}$ are shared rather than replicated at the three blocks. The learned gates combine the low-dimensional mixed streams, and the skip connection retains the incoming SAM feature. This construction transfers the prompt-space constraint to a spatially resolved residual adaptation path while avoiding four wide 1280-dimensional branches at every insertion site.

Operator-level rationale. Let $\mathcal{B}_n$ denote the set of nonnegative matrices whose row and column sums equal one. This constraint gives a useful norm bound on cross-stream mixing.

Proposition 1 (Non-expansive mixing). For $\mathbf{H} \in \mathcal{B}_n$, the Birkhoff–von Neumann decomposition gives $\mathbf{H} = \sum_k \omega_k \mathbf{\Pi}_k$, where $\mathbf{\Pi}_k$ are permutation matrices, $\omega_k \geq 0$, and $\sum_k \omega_k = 1$. Consequently,

$$\|\mathbf{H}\|_2 \leq \sum_k \omega_k \|\mathbf{\Pi}_k\|_2 = 1, \qquad \|\mathbf{HZ}\|_2 \leq \|\mathbf{Z}\|_2. \tag{8}$$

Row normalization alone does not provide this bound. For example, a matrix with every row equal to the first basis vector has unit row sums but spectral norm $\sqrt{n}$. Column normalization prevents this repeated allocation to a single input stream. This is the structural motivation for constraining the mixer before subsequent transformations.

Scope of the bound. The proposition concerns an exact doubly stochastic matrix, while finite Sinkhorn iterations approximate that condition. It does not bound the complete residual block or guarantee preservation of neighborhood distances. Moreover, the learned mixer depends on its input. For two inputs, the difference includes both $\mathbf{H}(Z_1)(Z_1 - Z_2)$ and $(\mathbf{H}(Z_1) - \mathbf{H}(Z_2))Z_2$; the proposition bounds only the first term. Feature distortion and threshold sensitivity therefore remain empirical questions, evaluated in Section 4.4.

3.4 Supporting Conditioning and Mask Selection

HyperCond and RankDice-RMA address conditioning and output selection around the constrained-fusion core (Fig. 5). HyperCond encodes the supplied threshold into a dense-prompt condition. RankDice-RMA selects a mask from ranked prediction probabilities. This functional separation permits evaluation of fusion choices and supporting components through distinct comparisons.

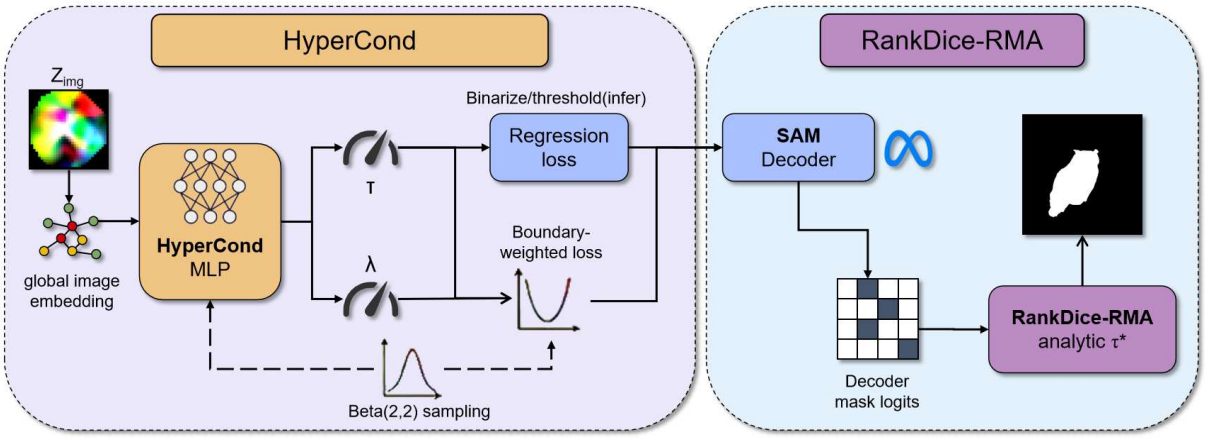


Fig. 5 HyperCond and RankDice-RMA perform distinct operations. Conditions enter before decoding. The selected threshold τ^* is used only for mask selection during inference. It is distinct from the condition input t and does not enter the training loss.

3.4.1 Hyperparameter Conditioning

HyperCond maps a scalar threshold condition t to a learned embedding:

$$\mathbf{c} = f_\phi(t) = \mathrm{MLP}(t), \qquad e^{\mathrm{dense}} = \mathrm{DenseHead}(F) + 0.1\psi(\mathbf{c}). \tag{9}$$

The projection ψ aligns and spatially expands the condition for dense prompting using a depthwise-separable spatial convolution. The boundary coefficient λ_b weights the training loss only; it is not an input to HyperCond.

During training, independent $\mathrm{Beta}(2,2)$ samples are rescaled to $t \in [0.4, 0.6]$ and $\lambda_b \in [0.3, 1.0]$. Threshold sampling exposes the model to nearby operating conditions, while boundary-weight sampling varies the strength of boundary supervision. At inference, the condition is fixed to $t = 0.5$ by default, and λ_b is absent. We reserve τ^* for the optional RankDice-RMA output threshold, distinguishing the supplied condition from the final mask decision.

3.4.2 RankDice-RMA: Optional Inference-Time Selection

RankDice-RMA is an optional inference-only post-processing step inspired by the ranking-based Dice selection of Dai and Li [7] and the subsequent reciprocal-moment approximation of Wang and Dai [51]. It introduces no trainable parameters and does not modify the training objective. Our implementation uses the simpler expected-cardinality surrogate below, rather than the exact reciprocal-moment solver of [51]; thus the module name denotes its methodological inspiration, not an implementation-equivalence claim. The selector maximizes this probability-based surrogate without using ground-truth labels. It does not guarantee maximization of the true Dice score. Let $\mathbf{S} = \{s_1, \ldots, s_N\}$ denote the predicted probability map. The ground-truth Dice score at threshold τ is

$$\mathrm{Dice}(\tau) = \frac{2\sum_{i=1}^{N} \mathbb{I}(s_i \geq \tau)\, y_i}{\sum_{i=1}^{N} \mathbb{I}(s_i \geq \tau) + \sum_{i=1}^{N} y_i}. \tag{10}$$

Because the ground-truth labels y_i are unavailable during inference, we adopt an expected-cardinality approximation. The procedure is summarized in Algorithm 1.

Algorithm 1 RankDice-RMA post-processing procedure

Require: Probability map $\mathbf{S} \in [0,1]^{H \times W}$

1: Flatten $\mathbf{S}$ into a vector containing N pixels
2: Sort the pixels in descending order: $\mathbf{s}_{\mathrm{sorted}} = \mathrm{Sort}(\mathbf{S}) = [s_{(1)}, \ldots, s_{(N)}]$
3: Calculate the cumulative sum: $S_k = \sum_{j=1}^{k} s_{(j)}$
4: Calculate the expected-cardinality surrogate π_{EC} for each rank k, where S_k approximates the intersection (true positives) and S_N serves as a proxy for the total object area:

$$\pi_{\mathrm{EC}}(k) = \frac{2S_k}{k + S_N}$$

5: Find the selected rank $k^* = \arg\max_k(\pi_{\mathrm{EC}}(k))$
6: Set the corresponding threshold $\tau^* = s_{(k^*)}$

Ensure: Binary mask $M = \mathbb{I}(\mathbf{S} \geq \tau^*)$

Sorting requires $O(N \log N)$ operations, followed by a linear scan over candidate ranks. The selected threshold adapts to the predicted probability distribution without using test labels. This selection operates on the probability map of a trained model. HyperCond supplies a prompt condition before decoding; RankDice-RMA selects an output threshold afterward.

The selector changes the binary decision $M = \mathbb{I}(\mathbf{S} \geq \tau^*)$, but leaves the underlying probability map $\mathbf{S}$ unchanged. Consequently, calibration error or a full threshold sweep computed

from **S** cannot improve solely because this selector is enabled. Such changes require a different probability map or a different evaluation object, which must be identified explicitly.

3.5 Dense Prompt Construction and Training

The dense head connects the adapted visual representation to the decoder. SAM combines image embeddings with sparse and dense prompts; without a mask prompt, its dense input uses a learned no-mask embedding. We instead derive an image-dependent dense embedding and add the HyperCond contribution.

To provide a semantic dense embedding, we derive e_i^{dense} from the image embedding e_i^{encoder} produced by the encoder of SAM, which inherently contains rich semantic information. Specifically, e_i^{encoder} is passed through two depthwise-separable convolutional stages coupled with GELU activations to generate e_i^{dense} (Fig. 6):

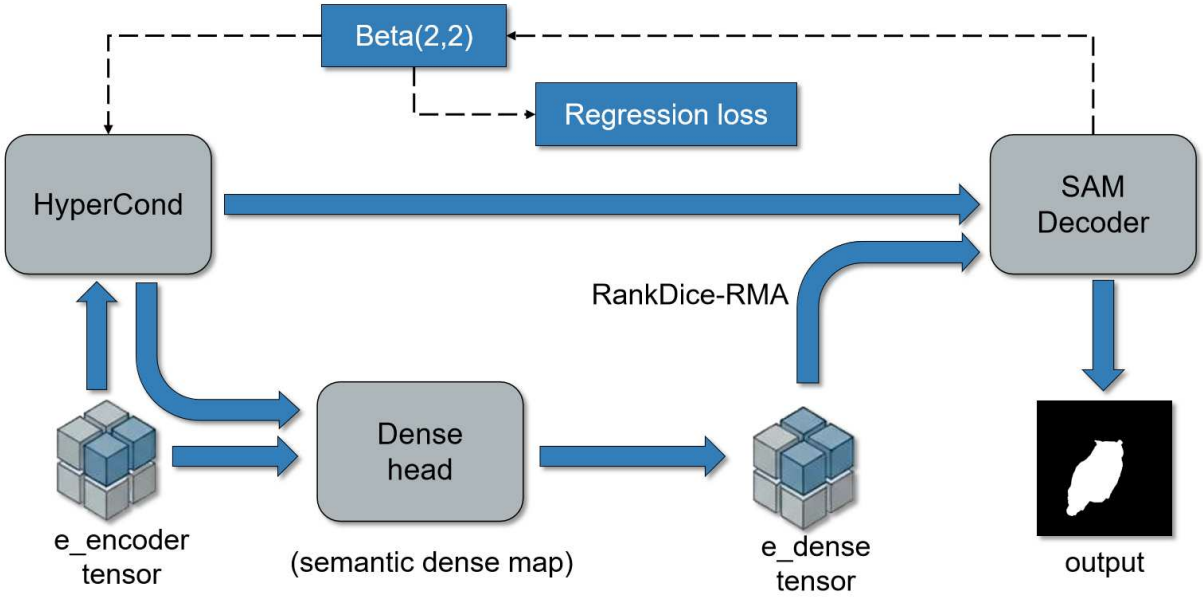


Fig. 6 Dense-prompt construction and decoding. The original schematic is retained. HyperCond supplies threshold conditioning to the dense prompt; λ_b weights training boundary supervision only. RankDice-RMA selects the final mask only at inference, as defined in the text.

$$e_i^{\text{dense}} = \text{GELU}\Big(\text{DSConv}\big(\text{GELU}(\text{DSConv}(e_i^{\text{encoder}}))\big)\Big) + 0.1\psi(\mathbf{c}), \tag{11}$$

where GELU denotes the Gaussian Error Linear Unit activation function and DSConv denotes a 3×3 depthwise convolution with padding 1 followed by a 1×1 pointwise projection. Both dense-head stages use 256 input and output channels. HyperCond maps the scalar threshold through widths $1 \to 64 \to 128$, then through $128 \to 256 \to 128$; a 1×1 projection maps the result to 256 channels, followed by a 256-channel depthwise-separable spatial projection. Following the design protocol of SAM, e_i^{encoder} is added to e_i^{dense}, and the resulting sum is fed into the mask decoder alongside e_i^{sparse} to predict the final mask S_{final}:

$$S_{\text{final}} = \text{Decoder}(e_i^{\text{dense}} + e_i^{\text{encoder}},\ e_i^{\text{sparse}}). \tag{12}$$

Training freezes the original SAM image-encoder layers and the BLIP and Mamba cue extractors. The trainable components include the single shared bottleneck MCA, rank-48 feature alignment, CSPG, depthwise-separable HyperCond and dense-head projections, and the SAM mask decoder. The decoder remains fully trainable in the reported configuration; freezing it

would reduce the trainable count but would not reduce the total model size. The segmentation objective combines Dice and binary cross-entropy terms with boundary supervision:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{Dice}} + \mathcal{L}_{\text{BCE}} + \lambda_b \mathcal{L}_{\text{boundary}}, \qquad \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}}. \tag{13}$$

Let B denote the soft boundary-weight map derived from the ground-truth mask G: a 3×3 morphological dilation-minus-erosion map is smoothed by 3×3 average pooling and normalized by its maximum. Let z denote the prediction logits. The boundary term is $\mathcal{L}_{\text{boundary}} = 0.05(\mathcal{L}_{\text{Dice},B} + \mathcal{L}_{\text{BCE},B})$ when $\sum_i B_i > 10$, and zero otherwise. Here $\mathcal{L}_{\text{Dice},B}$ computes Dice loss from $\text{sigmoid}(z)B$ and GB, while $\mathcal{L}_{\text{BCE},B}$ applies binary cross-entropy with logits to zB and GB, averaging over the full map. The base Dice loss uses squared predictions. The outer coefficient λ_b is applied once. RankDice-RMA is excluded from the training objective.

The experiments use SAM ViT-H. BLIP (`Salesforce/blip-image-captioning-large`) generates captions with its default generation configuration, and the MM-SAM-supplied Mamba text encoder (`state-spaces/mamba-130m-hf`) processes the captions. Both cue extractors remain frozen. In CSPG, the Gram-affinity temperature is 1.0 and Sinkhorn uses five row–column normalization pairs. MCA uses 20 pairs, four streams, and gates initialized to 0.01 before softmax. RankDice-RMA is applied only during inference, using the same trained weights as fixed-threshold prediction.

Images are resized to 1024×1024, converted to RGB, and normalized with ImageNet channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$; ground-truth masks use nearest-neighbor resizing. No random image augmentation is applied. We train for 20 epochs with batch size 1 on an NVIDIA RTX 5090 GPU with 32 GB of memory. AdamW uses an initial learning rate of 5×10^{-5}, weight decay 0.01, and cosine scheduling. A fixed 404-image validation subset is drawn from the combined 4,040-image COD10K/CAMO training pool with split seed 42, leaving 3,636 images for parameter updates. After each epoch, RankDice-RMA is enabled for validation inference, and we score the checkpoint by averaging validation S-measure, adaptive E-measure, and weighted F-measure. We retain the highest-scoring checkpoint across the 20 epochs. This checkpoint is then evaluated without RankDice-RMA for Table 2 and the probability-map ablation rows; RankDice-RMA is enabled only for the designated optional-inference comparisons. RankDice-RMA does not enter the training loss. Official test images are excluded from training and checkpoint selection and are evaluated only after model selection.

Table 1 gives the parameter accounting for the reported ViT-H configuration. It counts registered parameters in the integrated segmentation model, including the trainable SAM mask decoder and frozen SAM encoder and prompt encoder. The separately loaded frozen BLIP and Mamba models are excluded from the total. The shared MCA is counted once although it is applied at three encoder blocks.

4 Experiments

4.1 Experiment Setup

4.1.1 Dataset

We evaluate MC-SAM on four widely used COD benchmark datasets. COD10K [9], the largest COD benchmark, contains 5,066 images (3,040 training, 2,026 testing) across five super-classes and 69 sub-classes, providing matting-level annotations to facilitate learning fine object boundaries. CAMO [23] contains 1,250 images (1,000 training, 250 testing) of naturally and artificially camouflaged objects spanning eight categories. NC4K [36] contains 4,121 test images encompassing diverse natural and artificial camouflage scenarios. CHAMELEON [46] provides 76 internet-collected test images.

Following the standard protocol [5, 9], the source training pool comprises 4,040 images combining all training images from COD10K and CAMO. We reserve 404 of these images for

Table 1: Parameter counts by module for the reported ViT-H configuration. The integrated model has 642,183,656 parameters in total, of which 5,151,388 are trainable and 637,032,268 are frozen. Frozen BLIP and Mamba cue extractors are outside this count.

Module	Trainable	Frozen
Original SAM image encoder	0	637,026,048
SAM prompt encoder	0	6,220
Shared bottleneck MCA	580,276	0
BLIP feature alignment	99,584	0
CSPG	100,868	0
HyperCond	75,264	0
Conditional channel projection	32,768	0
Conditional spatial projection	68,096	0
Dense-prompt head	136,192	0
SAM mask decoder	4,058,340	0
Total	5,151,388	637,032,268

checkpoint validation and train on the remaining 3,636. Evaluation uses the complete NC4K and CHAMELEON datasets alongside the official CAMO and COD10K test splits.

The experiments address three questions:

- **Q1: Does MC-SAM improve camouflage segmentation?** Benchmark and qualitative comparisons establish overall performance.
- **Q2: What evidence supports constrained fusion?** Factorized ablations and stronger fusion comparisons examine segmentation scores alongside feature and output diagnostics.
- **Q3: What do the supporting components contribute?** Conditioning and threshold experiments examine gains beyond the fusion design.

MM-SAM [44] provides the closest multimodal SAM reference on COD10K, CAMO, and CHAMELEON. Table 2 uses its published results, whereas Tables 3 and 6 use our local reproduction. Its unreported NC4K results are shown as "–". Repeated-run results complement these comparisons by reporting variability.

4.1.2 Evaluation Metrics

We adopt four standard evaluation metrics: Structure-measure (S_α), mean E-measure (E_ϕ), weighted F-measure (F_β^w), and mean absolute error (MAE). Structure-measure (S_α) evaluates the structural similarity between the prediction and the ground truth by combining region-aware similarity (S_r) and object-aware similarity (S_o):

$$S_\alpha = \alpha S_o + (1-\alpha) S_r, \tag{14}$$

where $\alpha = 0.5$. E-measure (E_ϕ) captures both local pixel alignment and global image-level statistics through an alignment matrix ϕ:

$$E_\phi = \frac{1}{W \times H - 1 + \epsilon} \sum_{x=1}^{W} \sum_{y=1}^{H} \phi_{FM}(x, y). \tag{15}$$

Weighted F-measure (F_β^w) balances precision and recall while accounting for the relative importance of pixels:

$$F_\beta^w = \frac{(1+\beta^2) Precision^w \cdot Recall^w}{\beta^2 \cdot Precision^w + Recall^w}, \tag{16}$$

where $\beta^2 = 1$. Mean absolute error (MAE) measures the average pixel-wise absolute difference between the normalized prediction map $\hat{Y}$ and the binary ground truth G:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |\hat{Y}(x, y) - G(x, y)|. \tag{17}$$

The offline evaluator reads saved prediction maps, restores the ground-truth resolution when necessary, and normalizes the predictions before metric computation. For MC-SAM, the reported mean E-measure is the arithmetic mean of the per-image E-measure scores over the evaluated test set. The other scalar metrics are likewise computed per image and averaged over the test set. F-measure, E-measure, and precision–recall curves supplement the scalar results. The supplementary protocol, theoretical scope, and parameter accounting are provided in Online Resource 1 (ESM_1.pdf).

We distinguish the continuous probability map from the final binary mask. With RankDice-RMA disabled, the inference implementation exports $\mathbf{S} = \text{sigmoid}(z)$; enabling the selector exports $M = \mathbb{I}(\mathbf{S} \geq \tau^*)$. The main benchmark uses probability-map evaluation. The optional-selector rows evaluate the exported binary mask, so their score changes describe the chosen output representation, not improved probability calibration. HyperCond's input condition remains $t = 0.5$ for this comparison. A matched comparison of binary decisions would instead evaluate both $\mathbb{I}(\mathbf{S} \geq 0.5)$ and $\mathbb{I}(\mathbf{S} \geq \tau^*)$ on the same maps.

4.2 Q1: Overall Segmentation Performance

4.2.1 Overall competitiveness and closest external reference

Table 2 summarizes overall segmentation performance across four COD benchmarks and addresses Q1. The comparison places MC-SAM alongside task-specific segmentation methods and SAM-based adaptations, with MM-SAM serving as the closest multimodal SAM reference. Differences in model architectures and experimental settings make this table a benchmark comparison rather than a controlled test of the fusion mechanism.

On COD10K, MC-SAM achieves $S_\alpha = .903$, $E_\phi = .952$, $F_\beta^w = .858$, and MAE= .018. Relative to MM-SAM, these scores correspond to improvements of approximately 0.8%, 5.0%, and 6.2%, respectively, and a 21.7% reduction in MAE. The comparison with task-specific methods is more mixed. MC-SAM exceeds LPMoE in S-measure (.903 versus .896), while their weighted F-measures are both .858 at the reported precision. LPMoE records a higher E-measure (.955) and lower MAE (.015). Compared with MSRNet, MC-SAM achieves a higher weighted F-measure (.858 versus .852), but lower S-measure and E-measure and higher MAE. These results support competitive performance without indicating uniform superiority across the four metrics.

On NC4K, MC-SAM records $S_\alpha = .896$, $F_\beta^w = .871$, and MAE= .029. Its structural score is below those of MSRNet (.905) and LPMoE (.900), and its weighted F-measure is also lower. Its MAE is higher than those of MSRNet (.026) and LPMoE (.024). This comparison identifies remaining performance gaps despite the competitive results on COD10K.

On CAMO, MC-SAM achieves $S_\alpha = .872$, $E_\phi = .928$, $F_\beta^w = .850$, and MAE= .044. It improves upon MM-SAM in all four reported metrics, including weighted F-measure (.850 versus .782) and MAE (.044 versus .059). However, its MAE remains above LPMoE's .035, and its S-measure is lower (.872 versus .890). The gains over the multimodal SAM reference therefore do not imply an advantage over every task-specific method.

On CHAMELEON, MC-SAM obtains $S_\alpha = .930$, $F_\beta^w = .904$, and MAE= .015, compared with .923, .853, and .027 for MM-SAM. Its E-measure also increases from .946 to .964. MSRNet records higher $S_\alpha = .932$, $E_\phi = .978$, and $F_\beta^w = .908$; LPMoE also reaches $F_\beta^w = .908$. The relative advantage therefore remains metric-dependent on this benchmark.

Overall, the reported results show improvements over MM-SAM on the three datasets for which its results are available, together with competitive but mixed comparisons against task-

specific methods. MC-SAM achieves these results with 5.151M trainable parameters and a frozen SAM ViT-H backbone. This parameter count characterizes the adaptation budget rather than end-to-end computational cost. The following controlled studies examine the contributions of the fusion pathways and auxiliary components.

Table 2: Segmentation performance on four camouflaged object benchmarks (S_α, E_ϕ, F_β^w, MAE). The MC-SAM row evaluates continuous probability maps with fixed HyperCond condition $t = 0.5$ and without RankDice-RMA. **Bold** indicates the best performance in each column; blue indicates the second best. The symbol "–" denotes an unreported result. MC-SAM reports mean E-measure.

Method	COD10K				NC4K				CAMO				CHAMELEON			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$
Supervised Setting																
SINet [9]	.777	.865	.680	.043	.808	.871	.769	.058	.746	.804	.702	.092	.869	.891	.740	.044
PraNet [10]	.790	.861	.672	.045	.822	.877	.763	.059	.769	.825	.711	.094	.873	.905	.752	.039
NLDF [35]	.701	.819	.473	.059	.738	.846	.586	.083	.665	.790	.495	.123	.798	.893	.652	.063
PiCANet [32]	.696	.788	.415	.081	.758	.835	.570	.088	.701	.799	.510	.125	.765	.846	.552	.085
CPD [53]	.750	.853	.531	.053	.787	.866	.645	.072	.716	.796	.556	.113	.857	.923	.731	.048
EGNet [62]	.736	.854	.517	.061	.777	.864	.639	.075	.732	.820	.604	.109	.797	.884	.649	.065
PoolNet [31]	.740	.844	.506	.056	.785	.865	.635	.073	.730	.819	.575	.105	.845	.933	.690	.054
SCRN [54]	.789	.880	.575	.047	.830	.897	.698	.059	.779	.850	.643	.090	.876	.939	.741	.042
CSNet [11]	.778	.871	.569	.047	.750	.793	.603	.088	.771	.849	.641	.092	.856	.928	.718	.047
F3Net [52]	.739	.819	.544	.051	.780	.848	.656	.070	.711	.780	.564	.109	.848	.917	.744	.047
ITSD [65]	.767	.861	.557	.051	.811	.883	.679	.064	.750	.830	.610	.102	.814	.901	.662	.057
MINet [40]	.770	.859	.608	.042	.812	.887	.720	.056	.748	.838	.637	.090	.855	.937	.771	.036
SSAL [60]	.668	.789	.454	.066	.699	.812	.561	.093	.644	.780	.493	.126	.757	.856	.639	.071
UCNet [59]	.776	.867	.633	.042	.811	.886	.729	.055	.739	.820	.640	.094	.880	.941	.817	.036
PF-Net [37]	.800	.878	.701	.040	.829	.888	.785	.053	.782	.842	.747	.085	.879	.912	.768	.036
TINet [67]	.793	.862	.680	.043	.829	.880	.774	.055	.781	.836	.729	.087	.875	.908	.760	.038
UGTR [56]	.818	.853	.713	.035	.839	.874	.788	.052	.785	.823	.738	.086	.882	.914	.776	.033
SINetV2 [8]	.815	.888	.719	.037	.847	.903	.805	.048	.820	.882	.782	.071	.889	.924	.796	.028
MGL [58]	.814	.890	.666	.035	.833	.893	.739	.053	.775	.842	.673	.088	.893	.941	.812	.031
SLSR [36]	.804	.892	.673	.037	.840	.907	.766	.048	.787	.854	.696	.080	.890	.948	.822	.030
UJSC [24]	.809	.891	.684	.035	.842	.907	.771	.047	.800	.873	.728	.073	.891	.955	.833	.030
C2FNet [3]	.813	.891	.723	.036	.838	.898	.795	.049	.796	.854	.762	.080	.886	.920	.783	.031
BGNet [4]	.831	.901	.754	.033	.851	.907	.820	.044	.812	.870	.790	.073	.892	.927	.801	.027
OCENet [30]	.827	.894	.741	.033	.853	.903	.818	.045	.802	.852	.767	.081	.884	.918	.784	.032
ZoomNet [39]	.839	.888	.766	.029	.857	.896	.818	.044	.820	.878	.794	.066	.894	.926	.805	.026
DGNet [20]	.823	.897	.729	.033	.857	.912	.814	.043	.839	.901	.807	.057	.903	.938	.824	.021
HitNet [16]	.857	.919	.805	.027	.861	.910	.840	.042	.817	.875	.802	.071	.908	.944	.836	.019
VSCode [34]	.869	.931	.806	.024	.891	.935	.863	.032	.873	.926	.839	.047	.915	.952	.861	.018
RISNet [49]	.874	.931	.817	.025	.882	.925	.854	.037	.870	.923	.840	.050	.912	.949	.856	.019
HRTNet [63]	.872	.937	.813	.022	.893	.938	.869	.030	.873	.929	.855	.046	–	–	–	–
PRNet [17]	.874	.943	.799	.022	.891	.942	.848	.031	.872	.930	.831	.050	.914	.973	.874	.020
BCLNet [68]	.841	.919	.739	.028	.857	.920	.796	.042	.830	.897	.772	.064	.909	.974	.862	.023
DCIFNet [2]	.868	.936	.800	.023	.869	.927	.828	.038	.839	.908	.797	.055	.920	.973	.900	.018
SCNet [29]	.794	.896	.676	.036	.828	.910	.764	.048	.813	.903	.749	.068	.839	.945	.774	.035
LPMoE [47]	.896	.955	**.858**	**.015**	.900	.950	**.881**	**.024**	**.890**	**.949**	**.875**	**.035**	.919	.969	**.908**	.016
MSRNet [1]	**.907**	**.962**	.852	.016	**.905**	**.953**	.873	.026	.888	.942	.861	.040	**.932**	**.978**	**.908**	.017
SAM-based Supervised Fine-tuning																
CLIP Surgery+SAM [28]	.601	.672	.448	.187	–	–	–	–	.573	.666	.466	.206	.637	.710	.557	.180
GenSAM [14]	.775	.838	.681	.067	–	–	–	–	.719	.775	.659	.113	.764	.807	.680	.090
GroundingSAM [45]	.764	.813	.670	.085	–	–	–	–	.707	.753	.656	.157	.744	.776	.662	.122
ProMaC [15]	.805	.876	.716	.042	–	–	–	–	.767	.846	.725	.090	.833	.790	.899	.044
SAM-Adapter [5]	.883	.918	.801	.025	–	–	–	–	.847	.873	.765	.070	.896	.919	.824	.033
MM-SAM (reported) [44]	.896	.907	.808	.023	–	–	–	–	.863	.901	.782	.059	.923	.946	.853	.027
MC-SAM (Ours)	.903	.952	**.858**	.018	.896	.944	.871	.029	.872	.928	.850	.044	.930	.964	.904	**.015**

4.2.2 Curve comparison

Figure 7 reports F-measure, E-measure, and precision–recall curves across the four datasets. The curves complement the scalar metrics by showing how performance changes across decision thresholds. MC-SAM exhibits competitive precision–recall behavior, particularly on COD10K and CHAMELEON, alongside metric-dependent differences on NC4K and CAMO.

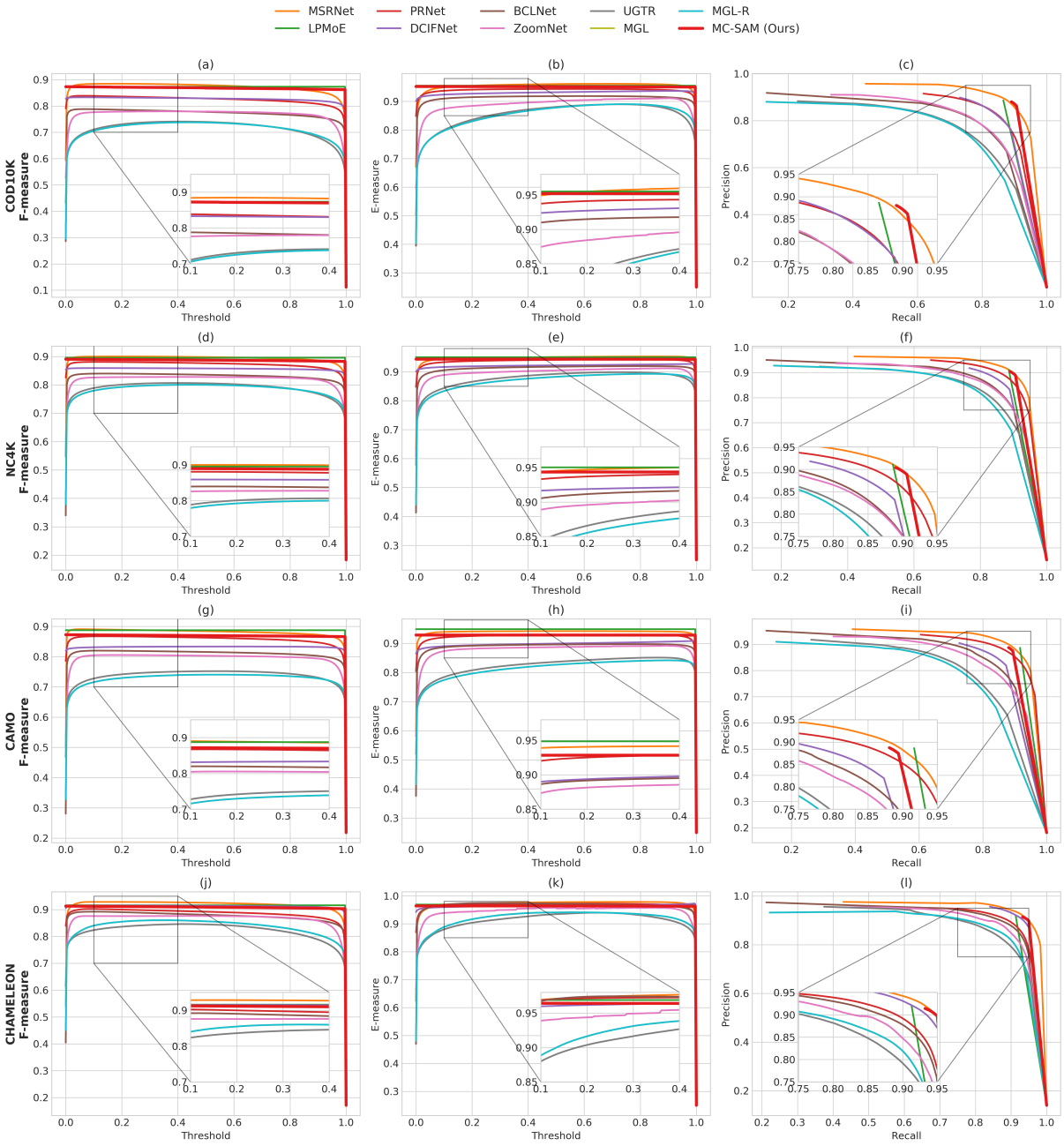


Fig. 7 F-measure, E-measure, and Precision-Recall (PR) curves of MC-SAM compared against representative state-of-the-art methods across the four COD benchmark datasets.

4.2.3 Qualitative evaluation

Figure 8 presents three examples from each dataset, covering textured backgrounds, fine structures, and partial occlusion. The displayed MC-SAM predictions retain coherent object regions and recover several thin structures that are fragmented in competing predictions.

The examples illustrate localization behavior across appearance, scale, and occlusion challenges. Component-level comparisons below examine how the adaptation pathways contribute to these overall results.

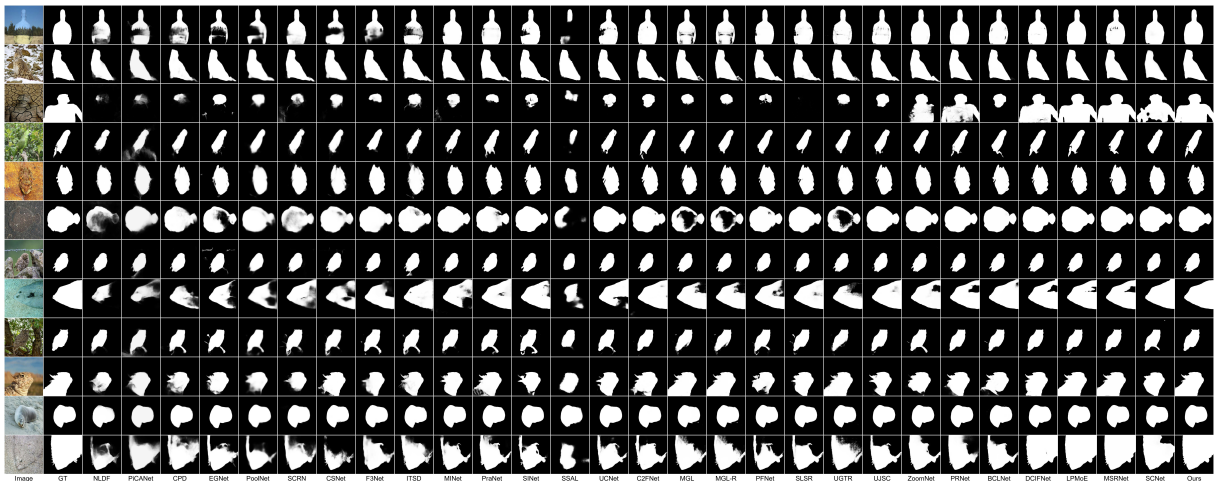


Fig. 8 Qualitative comparison on four COD datasets. Three examples per dataset illustrate textured backgrounds, fine structures, and occlusion. The displayed masks compare complete systems rather than individual adaptation modules.

4.3 Q2: Attributing Gains to the Fusion Core

Table 3: Factorized component study. Base denotes our reproduction of MM-SAM; its entries are means across independent training runs, with corresponding variability reported in Table 6. The first six variants evaluate continuous probability maps without RankDice-RMA; the last evaluates the binary mask selected from the trained complete model. For HC variants, the inference condition is fixed at $t = 0.5$. All variants use the segmentation objective, with no RankDice training loss. HC denotes HyperCond. Bold indicates the best value in each dataset column, including ties.

CAMO				
Variant	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	MAE↓
Base (our reproduction)	.863	.901	.782	.059
Base+CSPG	.864	.906	.795	.055
Base+MCA	.866	.915	.817	.050
Base+CSPG+MCA	.869	.921	.828	.047
Base+HC	.864	.905	.790	.055
MC-SAM (Base+HC+CSPG+MCA)	.872	.928	.850	.044
MC-SAM+RankDice-RMA	**.873**	**.930**	**.853**	**.043**
CHAMELEON				
Variant	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	MAE↓
Base (our reproduction)	.923	.946	.853	.027
Base+CSPG	.924	.951	.868	.023
Base+MCA	.927	.957	.882	.019
Base+CSPG+MCA	.928	.960	.892	.017
Base+HC	.924	.950	.865	.023
MC-SAM (Base+HC+CSPG+MCA)	.930	.964	.904	.015
MC-SAM+RankDice-RMA	**.931**	**.966**	**.907**	**.014**
COD10K				
Variant	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	MAE↓
Base (our reproduction)	.896	.907	.808	.023
Base+CSPG	.897	.919	.822	.021
Base+MCA	.899	.939	.842	.018
Base+CSPG+MCA	.901	.946	.849	.018
Base+HC	.901	.920	.820	.021
MC-SAM (Base+HC+CSPG+MCA)	.903	.952	.858	.018
MC-SAM+RankDice-RMA	**.904**	**.953**	**.861**	**.017**

Table 3 tests the fusion pathways before adding conditioning or output selection. Base de-

notes our locally reproduced MM-SAM configuration. Throughout the manuscript, MC-SAM denotes Base+HC+CSPG+MCA with RankDice-RMA disabled; MC-SAM+RankDice-RMA denotes the same trained network with the optional selector enabled. There is no additional trainable component between Base+HC+CSPG+MCA and MC-SAM.

4.3.1 Fusion gains without HyperCond

On COD10K, CSPG alone increases F_β^w from .808 to .822, while MCA alone increases it to .842. Combining CSPG and MCA raises F_β^w to .849, with $S_\alpha = .901$ and MAE= .018. CAMO weighted F-measure increases from .782 to .828, and CHAMELEON from .853 to .892. The core fusion pathways therefore improve segmentation without HC.

4.3.2 Supporting gains from conditioning and output selection

HC alone raises COD10K F_β^w to .820. Adding HC to the combined fusion core raises it from .849 to .858. The corresponding gains are .828 to .850 on CAMO and .892 to .904 on CHAMELEON. In this ablation sequence, the fusion core supplies the larger COD10K weighted F-measure increment, with HC adding a further improvement.

The optional-selector row reports COD10K $F_\beta^w = .861$ and MAE= .017 for its binary output, compared with .858 and .018 for the unselected probability map. The corresponding CAMO and CHAMELEON changes are also smaller than the fusion-core gains. The network and checkpoint are shared across these output modes; this comparison concerns post-processing and does not establish an improvement in the underlying probability estimates.

4.4 Q2: Matched Fusion-Operator Comparison

Table 4: Matched fusion comparisons in the joint COD10K setting. Variants share the backbone, training split, resolution, segmentation objective, and parameter budget. Entropic OT denotes transport mixing without the strict Birkhoff constraint. Bold indicates the best value in each column, including ties; the norm ratio is best when closest to 1.

(a) Segmentation

Fusion	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	MAE↓
Unconstrained	.900	.938	.842	.018
Row-softmax	.901	.945	.849	**.017**
Simple gated	.902	.947	.852	**.017**
Orthogonal	.902	.948	.853	**.017**
Spectral-normalized	.902	.949	.854	**.017**
Entropic OT	**.903**	.950	.855	**.017**
Sinkhorn-constrained	**.903**	**.952**	**.858**	.018

(b) Fusion diagnostics

Fusion	Dist.↓	Norm ratio	Logit var.↓	Sens.↓
Unconstrained	.25	1.26	.044	.026
Row-softmax	.17	1.15	.035	.019
Simple gated	.15	1.12	.032	.017
Orthogonal	.13	1.08	.029	.016
Spectral-normalized	.12	1.07	.028	.015
Entropic OT	.10	1.06	.026	.014
Sinkhorn-constrained	**.07**	**1.04**	**.024**	**.012**

Table 4 compares the joint fusion core against unconstrained, row-softmax, gated, orthogonal, spectral-normalized, and entropic OT alternatives. The matched setting tests whether the observed gains persist beyond simple fusion baselines.

Sinkhorn-constrained fusion obtains the highest COD10K weighted F-measure in this comparison, increasing it from .842 for unconstrained mixing to .858. Distance distortion decreases

from .25 to .07, norm ratio moves from 1.26 to 1.04, and logit variance decreases from .044 to .024. Threshold sensitivity decreases from .026 to .012.

The stronger alternatives narrow the segmentation gap. Spectral-normalized mixing reaches $F_\beta^w = .854$, and entropic OT reaches .855. Sinkhorn mixing also has lower distortion than entropic OT (.07 versus .10) and lower threshold sensitivity (.012 versus .014). These results support the row-and-column constraint beyond the use of transport-style smoothing alone.

The advantage remains metric-dependent: the other normalized alternatives obtain MAE= .017, compared with .018 for Sinkhorn. Taken together, the scores and diagnostics favor the constrained-fusion design on weighted F-measure, distortion, and threshold sensitivity, with a small MAE trade-off.

4.5 Q3: Conditioning and Output Selection

Table 5: HyperCond and output selection on COD10K. F/F and S/F evaluate continuous probability maps without RankDice-RMA. S/A uses the same trained network and inference condition as S/F, but evaluates the selected binary mask. The three scores therefore describe the specified output objects; S/F versus S/A is not a comparison of probability calibration. Bold identifies the highest reported output score.

Mode	HyperCond training	Evaluated output	$F_\beta^w \uparrow$
F/F	Fixed $t = 0.5$	Probability map $\mathbf{S}_\mathrm{F}$	.853
S/F	Sampled $t \in [0.4, 0.6]$	Probability map $\mathbf{S}_\mathrm{S}$	.858
S/A	Same training as S/F	Mask $\mathbb{I}(\mathbf{S}_\mathrm{S} \geq \tau^*)$	**.861**

Table 5 separates a training intervention from post-decoding output selection. F/F and S/F differ in exposure to HyperCond conditions during training. Both use the inference condition $t = 0.5$, and their weighted F-measure is evaluated on the exported probability map. Sampling the training condition increases the reported score from .853 to .858.

S/A reuses the S/F checkpoint and condition. RankDice-RMA selects τ^* from its probability map, and the resulting binary mask scores .861. This comparison measures the exported outputs, not an improvement in the probabilities that generated them. It also differs from a matched binary-mask comparison against a fixed decision threshold.

Calibration and threshold-sweep scope Let p denote a fixed continuous probability map, G the ground truth, $\mathcal{B}$ a fixed calibration-bin partition, and $\mathcal{T}$ a fixed threshold grid. Selecting τ^* leaves both $\mathrm{ECE}(p, G; \mathcal{B})$ and the curve $\{Q(\mathbb{I}(p \geq u), G) : u \in \mathcal{T}\}$ unchanged, for any fixed mask score Q. Its threshold-sweep area and the variance of per-image best thresholds are likewise unchanged. Recomputing these quantities on the binary output would change the evaluation object and would not measure calibration or threshold robustness of the original probability map. We therefore draw no probability-calibration conclusion from the S/F–S/A comparison.

4.6 Repeated-Run Variability

Table 6 reports between-run standard deviations of dataset-level scores; images within a test set are not counted as independent training repeats. The complete MC-SAM configuration has mean COD10K $F_\beta^w = .858$, consistent with Table 3. Its component list is exactly Base+HC+CSPG+MCA. The standard deviations describe run-to-run variability and are not confidence intervals. We use these summaries descriptively and make no significance claim from them.

Table 6: Descriptive variability across independent training runs, reported as mean±standard deviation. Base denotes our local MM-SAM reproduction. MC-SAM is Base+HC+CSPG+MCA, with no additional trainable component or RankDice-RMA selector, matching the complete-model row of Table 3. Base+HC+MCA excludes CSPG. All rows refer to probability-map evaluation. Bold identifies the highest mean.

(a) COD10K

Model	S_α	F_β^w
Base (MM-SAM, our reproduction)	.896 ± .0012	.808 ± .0028
Base+HC+MCA	.900 ± .0010	.850 ± .0022
MC-SAM (Base+HC+CSPG+MCA)	**.903 ± .0011**	**.858 ± .0021**

(b) Structural scores on CAMO and CHAMELEON

Model	CAMO S_α	CHAMELEON S_α
Base (MM-SAM, our reproduction)	.863 ± .0038	.923 ± .0055
Base+HC+MCA	.868 ± .0034	.928 ± .0046
MC-SAM (Base+HC+CSPG+MCA)	**.872 ± .0031**	**.930 ± .0041**

4.7 Computational Efficiency

Tables 7 and 8 report matched runtime and computational cost. Against locally run MM-SAM, MC-SAM without RankDice-RMA reduces latency by 91 ms/image, peak memory by 1.5 GB, and full-system cost by 0.21 TFLOPs/image. The optional selector adds 13 ms/image and 0.1 GB under this setup.

Table 7: Measured latency, throughput, and peak GPU memory for matched ViT-H MM-SAM and MC-SAM configurations. Both use BLIP and Mamba for text generation. RMA denotes the optional RankDice-RMA selector.

Model	Latency (ms/img)	Throughput (img/s)	Peak memory (GB)
MM-SAM (ViT-H)	563	1.8	10.5
MC-SAM (no RMA)	472	2.1	9.0
MC-SAM (+RMA)	485	2.06	9.1

Table 8: Parameters and FLOPs per image for the same configurations. The integrated parameter count excludes separately loaded frozen BLIP and Mamba cue extractors; full-system FLOPs include them. T denotes 10^{12} operations. Starred FLOPs exclude RMA post-processing.

Model	Trainable parameters	Integrated parameters	Full FLOPs (T/img)	SAM+adapters FLOPs (T/img)
MM-SAM (ViT-H)	9,164,932	646,190,980	3.45	3.02
MC-SAM (no RMA)	5,151,388	642,183,656	3.24	2.77
MC-SAM (+RMA)	5,151,388	642,183,656	3.24*	2.77*

4.8 Discussion

The central result is that constraining fusion improves the use of existing cues at both SAM adaptation interfaces. On COD10K, CSPG and MCA raise weighted F-measure by .041 before HyperCond adds a further .009 in the factorized sequence. The matched operator comparison provides complementary evidence: the Sinkhorn constraint improves weighted F-measure alongside lower feature distortion and threshold sensitivity. Together, these experiments support the fusion design through both component removal and substitution of the mixing operator.

The distinction between row and column constraints explains the design motivation. Row normalization controls each output mixture; column normalization also limits aggregate input

contribution. The operator bound formalizes this distinction for exact doubly stochastic mixing. The empirical diagnostics then examine whether the proposed design is accompanied by favorable representation and output behavior.

HyperCond and RankDice-RMA operate at different stages of this design. HyperCond changes the dense prompt before decoding and can therefore change the predicted probability map. RankDice-RMA acts after decoding and changes only the binary mask selected from that map. Its output scores provide evidence about mask selection; they cannot establish an improvement in probability calibration.

The evidence has several boundaries. The factorized ablations isolate fusion gains without HC, but concurrent diagnostic improvements do not identify a unique causal pathway. The operator bound has the scope stated in Section 3.3. Performance also depends on the metric, including the COD10K MAE trade-off. Weak cues, small objects, and occlusion remain relevant failure factors; CHAMELEON's 76 images also limit the breadth of that benchmark alone.

Future work should characterize which camouflage conditions benefit most from the constraint and examine its behavior under distribution shift. Additional analyses could separate the effects of branch allocation, feature scale, and threshold sensitivity. Efficiency comparisons should include cue extraction, adaptation, and output selection.

5 Conclusion

MC-SAM makes fusion stability an explicit design objective for SAM adaptation to camouflaged scenes. CSPG and a weight-shared bottleneck MCA impose the same doubly stochastic prior at the prompt and feature interfaces. Factorized ablations show gains before auxiliary conditioning or output selection, and matched fusion comparisons connect higher weighted F-measure with lower distortion and threshold sensitivity. The framework trains 5.151M parameters with a frozen SAM ViT-H image encoder. HyperCond conditions the dense prompt, while optional RankDice-RMA selects a binary mask from the decoder probabilities. The results support preserving weak foreground evidence through constrained fusion as a practical adaptation principle. Its theoretical guarantee remains operator-level; the empirical evidence does not establish fusion instability as the sole source of segmentation error.

Statements and Declarations

Funding. This research received support from the Guangdong Provincial Science and Technology Program for Regional Innovation Capability and Support System Construction, Guangdong Ocean University 2024 "Chongbuqiang" Provincial Financial Special Project (Grant No. 080503112401); the Ministry of Science and Technology High-end Foreign Experts Introduction Program (Grant No. G2023030082L); the Graduate Education Innovation Program of Guangdong Province (Document No. Yue Jiaoyan Han [2023] 3); and the Guangdong Provincial Science and Technology Program for Regional Innovation Capability and Support System Construction, Guangdong Ocean University 2025 "Chongbuqiang" Provincial Financial Special Project (Grant No. 080507202501).

Competing interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability. The COD10K, CAMO, NC4K, and CHAMELEON datasets are available from their original providers, as cited in Section 4.1.1. The MC-SAM training, inference, and parameter-counting source code is publicly available at https://github.com/l

ue2290/MC-SAM (revision 9035e90). Dataset images and pretrained or trained model weights are not redistributed in that repository.

References

[1] Alghamdi, L., Usman, M., Anwar, H., Bais, A., Anwar, S.: MSRNet: A multi-scale recursive network for camouflaged object detection. arXiv preprint arXiv:2511.12810 (2025). https://doi.org/10.48550/arXiv.2511.12810

[2] Cao, Y., Wang, D.: DCIFNet: Dual-branch cross-layer information flow network for camouflaged object detection in complex scenes. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) (2025). https://doi.org/10.1145/3764866

[3] Chen, G., Liu, S.J., Sun, Y.J., Ji, G.P., Wu, Y.F., Zhou, T.: Camouflaged object detection via context-aware cross-level fusion. IEEE Trans. Circuits Syst. Video Technol. **32**(10), 6981–6993 (2022). https://doi.org/10.1109/TCSVT.2022.3178173

[4] Chen, T., Xiao, J., Hu, X., Zhang, G., Wang, S.: Boundary-guided network for camouflaged object detection. Knowl.-Based Syst. **248**, 108901 (2022). https://doi.org/10.1016/j.knosys.2022.108901

[5] Chen, T., Zhu, L., Deng, C., Cao, R., Wang, Y., Zhang, S., Li, Z., Sun, L., Zang, Y., Mao, P.: SAM-adapter: Adapting segment anything in underperformed scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 3367–3375 (2023). https://doi.org/10.1109/ICCVW60793.2023.00361

[6] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in neural information processing systems. vol. 26, pp. 2292–2300 (2013). https://doi.org/10.5555/2999792.2999925

[7] Dai, B., Li, C.: RankSEG: A consistent ranking-based framework for segmentation. Journal of Machine Learning Research **24**(224), 1–50 (2023), https://www.jmlr.org/papers/v24/22-0712.html

[8] Fan, D.P., Ji, G.P., Cheng, M.M., Shao, L.: Concealed object detection. IEEE Trans. Pattern Anal. Mach. Intell. **44**(10), 6024–6042 (2021). https://doi.org/10.1109/TPAMI.2021.3085766

[9] Fan, D.P., Ji, G.P., Sun, G., Cheng, M.M., Shen, J., Shao, L.: Camouflaged object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2777–2787 (2020). https://doi.org/10.1109/CVPR42600.2020.00285

[10] Fan, D.P., Ji, G.P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: PraNet: parallel reverse attention network for polyp segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 263–273. Springer (2020). https://doi.org/10.1007/978-3-030-59725-2_26

[11] Gao, S., Tan, Y.q., Bian, J., Zhang, X., Zhang, J., Cheng, M.M., Yan, S.: Highly efficient salient object detection with 100k parameters. In: ECCV (2020). https://doi.org/10.1007/978-3-030-58539-6_42

[12] Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: Proceedings of the First Conference on Language Modeling (COLM) (2024), https://openreview.net/forum?id=tEYskw1VY2

[13] He, C., Li, K., Zhang, Y., Tang, L., Zhang, Y., Guo, Z., Li, X.: Camouflaged object detection with feature decomposition and edge reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22046–22055 (2023). https://doi.org/10.1109/CVPR52729.2023.02111

[14] Hu, J., Lin, J., Gong, S., Cai, W.: Relax image-specific prompt requirement in SAM: A single generic prompt for segmenting camouflaged objects. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 12511–12518 (2024). https://doi.org/10.1609/aaai.v38i11.29144

[15] Hu, J., Lin, J., Yan, J., Gong, S.: Leveraging hallucinations to reduce manual prompt dependency in promptable segmentation. In: Advances in Neural Information Processing Systems. vol. 37 (2024). https://doi.org/10.52202/079017-3403
[16] Hu, X., Wang, S., Qin, X., Dai, H., Ren, W., Luo, D., Tai, Y., Shao, L.: High-resolution iterative feedback network for camouflaged object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 881–889 (2023). https://doi.org/10.1609/aaai.v37i1.25167
[17] Hu, X., Zhang, X., Wang, F., Sun, J., Sun, F.: Efficient camouflaged object detection network based on global localization perception and local guidance refinement (prnet). IEEE Transactions on Circuits and Systems for Video Technology (2024). https://doi.org/10.1109/TCSVT.2023.3349209
[18] Huang, Z., Dai, H., Xiang, T.Z., Wang, S., Chen, H.X., Qin, J., Xiong, H.: Feature shrinkage pyramid for camouflaged object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5557–5566 (2023). https://doi.org/10.1109/CVPR52729.2023.00538
[19] Jeong, J., Shim, J., Yoon, H.: TANet: tri-aspects network for camouflaged object detection. IEEE Trans. Circuits Syst. Video Technol. **35**(11), 10728–10741 (2025). https://doi.org/10.1109/TCSVT.2025.3574144
[20] Ji, G.P., Fan, D.P., Chou, Y.C., Dai, D., Liniger, A., Van Gool, L.: Deep gradient learning for efficient camouflaged object detection. Mach. Intell. Res. **20**(1), 92–108 (2023). https://doi.org/10.1007/s11633-022-1365-9
[21] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023). https://doi.org/10.1109/ICCV51070.2023.00371
[22] Lai, Z., et al.: SCOUT: Semi-supervised camouflaged object detection by utilizing text and adaptive data selection. In: International Joint Conference on Artificial Intelligence (IJCAI) (2025)
[23] Le, T.N., Nguyen, T., Nie, Z., Tran, M.T., Sugimoto, A.: Anabranch network for camouflaged object segmentation. Comput. Vis. Image Underst. **184**, 45–56 (2019). https://doi.org/10.1016/j.cviu.2019.04.006
[24] Li, A., Zhang, J., Lv, Y., Liu, B., Zhang, T., Dai, Y.: Uncertainty-aware joint salient object and camouflaged object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10071–10081 (2021). https://doi.org/10.1109/CVPR46437.2021.00994
[25] Li, B., Xiao, H., Tang, L.: ASAM: Boosting segment anything model with adversarial tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3699–3710 (2024). https://doi.org/10.1109/CVPR52733.2024.00354
[26] Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 12888–12900 (2022)
[27] Li, Y., Wang, H., Duan, Y., Zhang, J., Li, X.: A closer look at the explainability of contrastive language-image pre-training. Pattern Recognition **162**, 111409 (2025). https://doi.org/10.1016/j.patcog.2025.111409
[28] Li, Y., Wang, H., Duan, Y., Zhang, J., Li, X.: A closer look at the explainability of contrastive language-image pre-training. Pattern Recognition **162**, 111409 (2025). https://doi.org/10.1016/j.patcog.2025.111409
[29] Li, Z., Zhang, D., Wang, C., et al.: SCNet: Adaptive context mining for camouflaged object detection with scribble supervision. Computer Vision and Image Understanding (CVIU) **259**, 104430 (2025). https://doi.org/10.1016/j.cviu.2025.104430

[30] Liu, J., Zhang, J., Barnes, N.: Modeling aleatoric uncertainty for camouflaged object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1445–1454 (2022). https://doi.org/10.1109/WACV51458.2022.00267
[31] Liu, J.J., Hou, Q., Cheng, M.M., Feng, J., Jiang, J.: A simple pooling-based design for real-time salient object detection. In: CVPR (2019). https://doi.org/10.1109/CVPR.2019.00404
[32] Liu, N., Han, J., Yang, M.H.: Picanet: Learning pixel-wise contextual attention for saliency detection. In: CVPR (2018). https://doi.org/10.1109/CVPR.2018.00327
[33] Liu, Y., et al.: Improving SAM for camouflaged object detection via dual stream adapters. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
[34] Luo, Z., Liu, N., Zhao, W., Yang, X., Zhang, D., Fan, D.P., Khan, F., Han, J.: Vscode: general visual salient and camouflaged object detection with 2d prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17169–17180 (2024). https://doi.org/10.1109/CVPR52733.2024.01626
[35] Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: CVPR (2017). https://doi.org/10.1109/CVPR.2017.698
[36] Lv, Y., Zhang, J., Dai, Y., Li, A., Liu, B., Barnes, N., Fan, D.P.: Simultaneously localize, segment and rank the camouflaged objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11591–11601 (2021). https://doi.org/10.1109/CVPR46437.2021.01142
[37] Mei, H., Ji, G.P., Wei, Z., Yang, X., Wei, X., Fan, D.P.: Camouflaged object segmentation with distraction mining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8772–8781 (2021). https://doi.org/10.1109/CVPR46437.2021.00866
[38] Pan, Y., Chen, Y., Fu, Q., Zhang, P., Xu, X., et al.: Study on the camouflaged target detection method based on 3d convexity. Mod. Appl. Sci. **5**(4), 152 (2011). https://doi.org/10.5539/mas.v5n4p152
[39] Pang, Y., Zhao, X., Xiang, T.Z., Zhang, L., Lu, H.: 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. pp. 2160–2170 (2022). https://doi.org/10.1109/CVPR52688.2022.00220
[40] Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: CVPR (2020). https://doi.org/10.1109/CVPR42600.2020.00943
[41] Pare, S., Bhandari, A., Kumar, A., Singh, G.: An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix. Expert Syst. Appl. **87**, 335–362 (2017). https://doi.org/10.1016/j.eswa.2017.06.021
[42] Pei, J., Cheng, T., Fan, D.P., Tang, H., Chen, C., Van Gool, L.: OSFormer: one-stage camouflaged instance segmentation with transformers. In: European Conference on Computer Vision. pp. 19–37. Springer (2022). https://doi.org/10.1007/978-3-031-19797-0_2
[43] Radford, A., Kim, J., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763 (2021)
[44] Ren, G., Liu, H., Lazarou, M., Stathaki, T.: MM-SAM: Multi-modal segment anything model for camouflaged scene segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19882–19892 (2025)
[45] Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded SAM: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024). https://doi.org/10.48550/arXiv.2401.14159
[46] Skurowski, P., Abdulameer, H., Błaszczyk, J., Depta, T., Kornacki, A., Kozieł, P.: Animal

camouflage analysis: Chameleon database. Unpublished manuscript (2018). https://doi.org/10.13140/RG.2.2.31306.98246
[47] Sun, Y., Lian, J., Yang, J., Luo, L.: Controllable-LPMoE: Adapting to challenging object segmentation via dynamic local priors from mixture-of-experts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22327–22337 (2025). https://doi.org/10.1109/ICCV51701.2025.02073
[48] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
[49] Wang, L., Yang, J., Zhang, Y., Wang, F., Zheng, F.: Depth-aware concealed crop detection in dense agricultural scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17201–17211 (2024). https://doi.org/10.1109/CVPR52733.2024.01629
[50] Wang, Q., et al.: Weakly supervised camouflaged object detection based on the SAM model and mask guidance. Image and Vision Computing (2025)
[51] Wang, Z., Dai, B.: RankSEG-RMA: An efficient segmentation algorithm via reciprocal moment approximation. In: Advances in Neural Information Processing Systems. vol. 38 (2025). https://doi.org/10.52202/085713-5452
[52] Wei, J., Wang, S., Huang, Q.: F3net: Fusion, feedback and focus for salient object detection. In: AAAI (2020). https://doi.org/10.1609/aaai.v34i07.6916
[53] Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: CVPR (2019). https://doi.org/10.1109/CVPR.2019.00403
[54] Wu, Z., Su, L., Huang, Q.: Stacked cross refinement network for edge-aware salient object detection. In: ICCV (2019). https://doi.org/10.1109/ICCV.2019.00736
[55] Xiao, J., et al.: A survey of camouflaged object detection and beyond. CAAI Artificial Intelligence Research (2024)
[56] Yang, F., Zhai, Q., Li, X., Huang, R., Luo, A., Cheng, H., Fan, D.P.: Uncertainty-guided transformer reasoning for camouflaged object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4146–4155 (2021). https://doi.org/10.1109/ICCV48922.2021.00411
[57] Yin, B., Zhang, X., Fan, D.P., Jiao, S., Cheng, M.M., Van Gool, L., Hou, Q.: CamoFormer: masked separable attention for camouflaged object detection. IEEE Trans. Pattern Anal. Mach. Intell. **46**, 10362–10374 (2024). https://doi.org/10.1109/TPAMI.2024.3442929
[58] Zhai, Q., Li, X., Yang, F., Jiao, Z., Luo, P., Cheng, H., Liu, Z.: Mutual graph learning for camouflaged object detection. IEEE Transactions on Image Processing (2021). https://doi.org/10.1109/TIP.2021.3058783
[59] Zhang, J., Fan, D.P., Dai, Y., Anwar, S., Saleh, F.S., Barnes, N.: Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In: CVPR (2020). https://doi.org/10.1109/CVPR42600.2020.00861
[60] Zhang, J., Yu, X., Dai, Y., Li, J., Van De Weijer, J.: Weakly-supervised salient object detection via scribble annotations. In: CVPR (2020). https://doi.org/10.1109/CVPR42600.2020.01256
[61] Zhang, M., Xu, S., Piao, Y., Shi, D., Lin, S., Lu, H.: PreyNet: preying on camouflaged objects. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 5323–5332 (2022). https://doi.org/10.1145/3503161.3548178
[62] Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet: Edge guidance network for salient object detection. In: ICCV (2019). https://doi.org/10.1109/ICCV.2019.00887
[63] Zhao, Y., et al.: HRTNet: Holistic registration theory-inspired network for camouflaged object detection. Neurocomputing (2026). https://doi.org/10.1016/j.neucom.2026.132674
[64] Zheng, D., Zheng, X., Yang, L., Gao, Y., Zhu, C., Ruan, Y.: MFFN: multi-view feature

fusion network for camouflaged object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6232–6242 (2023). https://doi.org/10.1109/WACV56688.2023.00617

[65] Zhou, H., Xie, X., Lai, J.H., Chen, Z., Yang, L.: Interactive two-stream decoder for accurate and fast saliency detection. In: CVPR (2020). https://doi.org/10.1109/CVPR42600.2020.00916

[66] Zhou, X., Wu, Z., Cong, R.: Decoupling and integration network for camouflaged object detection. IEEE Trans. Multimed. **26**, 7114–7129 (2024). https://doi.org/10.1109/TMM.2024.3360710

[67] Zhu, J., Zhang, X., Zhang, S., Liu, J.: Inferring camouflaged objects by texture-aware interactive guidance network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3599–3607 (2021). https://doi.org/10.1609/aaai.v35i4.16475

[68] Zhuang, C., Zhang, Q., Zhang, C., Yuan, X.: Boundary-and-object collaborative learning network for camouflaged object detection. Image and Vision Computing **161**, 105596 (2025). https://doi.org/10.1016/j.imavis.2025.105596

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ESM1.pdf