PoEM: Reference-Guided Camouflaged Object Detection via Probabilistic Environment Modeling

Anonymous authors
Paper under double-blind review

Abstract

Reference-guided camouflaged object detection (RGCOD) aims to localize camouflaged objects in a query image using reference information. Existing approaches predominantly formulate RGCOD as feature matching: text-based methods are constrained by the cross-modal semantic gap, whereas vision-based methods typically rely on paired images with pixel-level annotations, limiting scalability and generalization. We propose **PoEM**, a training-free framework that recasts RGCOD as Probabilistic Environment Modeling, requiring only target-free reference environment images and no paired annotations. The central idea is to estimate the query image's background distribution from reference environments and identify camouflaged objects as regions that deviate from this distribution. To this end, we introduce a Describe–Sample–Filter (DSF) agent that automatically constructs high-quality reference environment images that are semantically and structurally aligned with the query image. We then use probabilistic principal component analysis (PPCA) to model the reference environment distribution in DINOv3 feature space and score each query patch by its negative log-likelihood under the resulting model. Higher scores indicate lower compatibility with the reference environment, providing cues for camouflaged object localization. When integrated with the Segment Anything Model (SAM), PoEM consistently outperforms state-of-the-art SAM-based RGCOD methods across multiple benchmarks.

1 Introduction

Camouflaged object detection (COD) (Fan et al., 2020) aims to segment targets that blend into their backgrounds. Although supervised methods (Fan et al., 2022; Pang et al., 2022; He et al., 2023a; 2026) have made substantial progress, they are typically designed for predefined, closed-set settings. These settings may not fully capture the complexity of real-world environments, limiting generalization to unseen scenes.

To address this limitation, reference-guided camouflaged object detection (RGCOD) (Hu et al., 2024b; Du et al., 2026) uses limited reference information to guide the detection of camouflaged targets in a query image. The central challenge is to establish effective interactions between the query image and the reference information. Most existing approaches rely on feature matching to enable these interactions. Based on the reference modality, these methods can be categorized as text-based or vision-based, as illustrated in Fig. 1.

Text-based methods (Hu et al., 2024a; Du et al., 2025c) are often limited by the modality gap between textual semantics and visual cues, whereas vision-based methods (Nguyen et al., 2024; Du et al., 2026) typically require reference images paired with target annotations or pseudo-annotations. Moreover, both paradigms are built on a feature-matching assumption: the reference information should contain semantic content that is at least partially similar to the query target. Whether the guidance is text-based or vision-based, obtaining reference information that is accurately aligned with target semantics at low cost remains the main bottleneck of this line of research.

The environment-centric retrieval strategy introduced by the unsupervised method EASE (Du et al., 2025a) offers a new perspective on addressing this limitation. However, direct retrieval matches features against discrete prototypes without explicitly modeling the overall background distribution

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

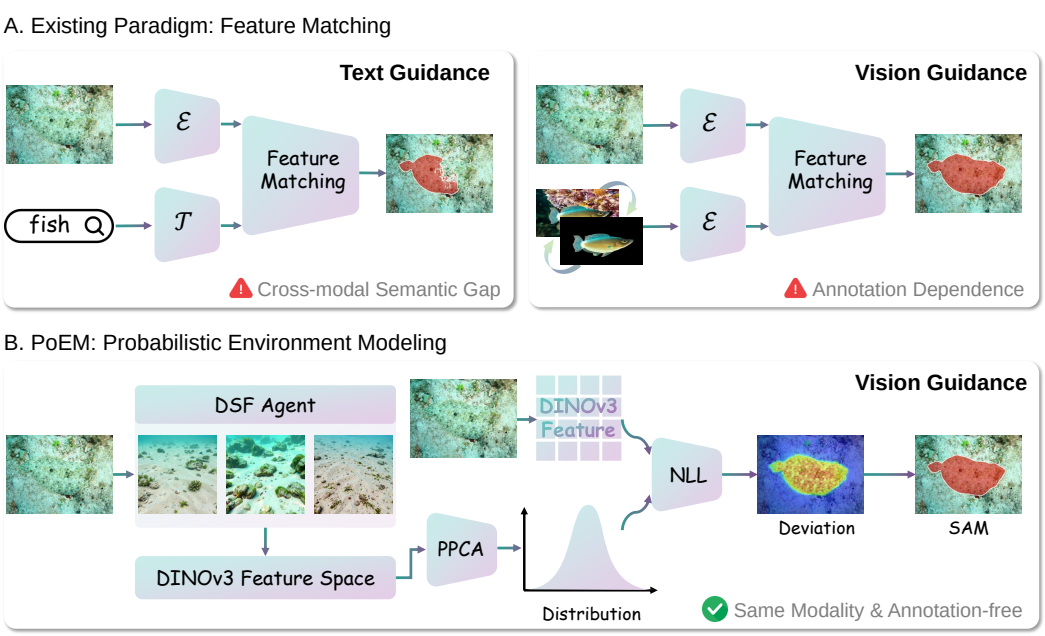


Figure 1: Comparison between existing methods and PoEM. In contrast to existing feature-matching paradigms, PoEM revisits RGCOD from the perspective of probabilistic environmental modeling and relies solely on reference environment images, without requiring any target-related reference information.

or its scale of variation, making it sensitive to limited prototype coverage and noise. To address these limitations, we move beyond the feature-matching paradigm in RGCOD and introduce PoEM, a probabilistic environment modeling framework for target-free reference guidance. Specifically, PoEM models only the environment of the query image as a probability distribution and uses features that deviate from this distribution as cues for localizing candidate camouflaged regions. This focus on the environment allows PoEM to use target-free reference images, eliminating the need for target annotations in the reference set.

Specifically, we introduce a Describe-Sample-Filter (DSF) agent to automatically acquire target-free reference images whose backgrounds closely match that of the query image. In the DINOv3 (Siméoni et al., 2025) feature space, we fit a probabilistic principal component analysis (PPCA) (Tipping & Bishop, 1999) model to the reference features and score each query feature using its negative log-likelihood (NLL) under this model. Higher scores indicate greater deviations from the modeled environment distribution, yielding a deviation map that highlights candidate camouflaged regions. We extract prompts from this map to guide the Segment Anything Model (SAM) (Ravi et al., 2025) in producing refined segmentation masks. To further improve environment modeling, we incorporate automatic selection of the number of principal components based on the Bayesian information criterion (BIC), along with a multi-layer feature representation.

PoEM combines probabilistic environment modeling with the generalization capabilities of foundation models to enable training-free camouflaged object detection. Extensive experiments demonstrate that PoEM outperforms other SAM-based methods across multiple benchmarks.

Our contributions are summarized as follows:

• We introduce PoEM, a probabilistic environment modeling framework for RGCOD that moves beyond feature matching. By relying exclusively on target-free reference images, PoEM relaxes the requirement for target-level semantic alignment between reference and query images, offering a practical path toward scalable and generalizable RGCOD.

• We propose the Describe-Sample-Filter (DSF) agent, which uses the conditional generation capabilities of foundation models to automatically produce target-free reference images with backgrounds closely aligned with that of the query image.

• We develop a probabilistic pipeline that integrates PPCA-based environment modeling, BIC-based principal component selection, and multi-layer feature representations to produce deviation scores for camouflaged object localization. Experiments demonstrate that PoEM achieves competitive performance across multiple benchmarks.

2 Related Work

2.1 Reference-Guided Camouflaged Object Detection

Unlike mainstream COD research, which primarily studies how to learn discriminative representations of camouflaged targets under different annotation regimes, including fully supervised (Fan et al., 2020; Pang et al., 2022; He et al., 2024; 2025b; Hao et al., 2025), weakly supervised (He et al., 2023c; 2025a), and unsupervised settings (Du et al., 2025b; Liu et al., 2026; Chen et al., 2026b), reference-guided camouflaged object detection (RGCOD) (Hu et al., 2024a;b; Du et al., 2025c; 2026) focuses on how to exploit limited reference information to narrow the search space and facilitate target localization. Compared with supervised methods that mainly target closed-set scenarios, RGCOD is better suited to flexible search in open-world settings based on reference priors.

Most existing RGCOD methods follow a feature-matching paradigm to establish interactions between the query image and reference information. According to the reference modality, these methods can be broadly categorized into text-based and vision-based approaches. Text-based methods (Hu et al., 2024a;b; Yin et al., 2025; Du et al., 2025c) typically rely on pretrained vision-language models to extract cross-modal features and perform matching via cross-attention mechanisms. Vision-based methods (Du et al., 2026; Zhang et al., 2025) alleviate the modality gap inherent in text-based approaches and provide more direct visual priors. These methods usually assume the availability of paired reference images with annotations. Their core interaction mechanism is also primarily based on feature matching with cross-attention.

PoEM is a vision-based method. Unlike prior approaches that rely on feature matching for target retrieval, it formulates reference guidance through probabilistic environment modeling. This formulation requires only environment images as references, eliminating the need for paired reference images and annotations. By relaxing existing constraints on the reference format, PoEM offers a new perspective on problem formulation and method design in RGCOD.

2.2 Foundation-Model-Driven Segmentation

Pretrained on large-scale data corpora, foundation models have demonstrated strong generalization across various downstream segmentation tasks (Kirillov et al., 2023; Oquab et al., 2024). Despite these advantages, applying foundation models to camouflaged object segmentation remains challenging due to the distinctive characteristics of COD. Existing methods, whether zero-shot approaches (Tang et al., 2024; Yang et al., 2026) based on SAM or unsupervised methods centered on DINO (Yan et al., 2025; Jiang et al., 2026), still largely treat camouflaged object segmentation as a natural extension of generic object segmentation. In highly challenging cases, however, this implicit assumption may limit the extent to which foundation models realize their generalization potential. The unsupervised method EASE (Du et al., 2025a) offers a different perspective on COD in the era of foundation models by replacing object discovery with environment retrieval. Motivated by this perspective, we further introduce the environment-centered idea into RGCOD and exploit reference environment information through a probabilistic modeling framework. This paradigm emphasizes environment modeling rather than explicit feature matching, which allows foundation models to focus on the salient representations for which they are naturally well suited, while addressing the core difficulties of COD through the complementary strengths of different model families.

3 Method

The proposed PoEM addresses reference-guided camouflaged object detection through a probabilistic environment modeling framework. The model takes a query image and several reference environment images as input and outputs the mask of the camouflaged target. An overview of PoEM is shown in Fig. 2. We first introduce the DSF agent that automatically creates reference environment

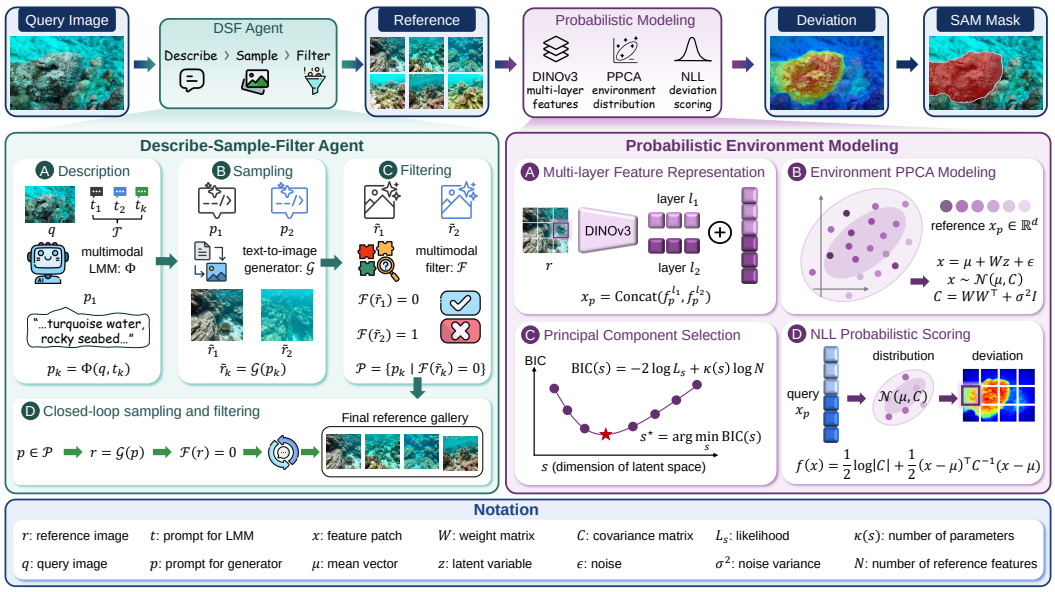


Figure 2: Overview of the proposed PoEM.

images for each query image. We then present a probabilistic modeling framework to segment camouflaged objects, including PPCA-based probabilistic modeling, BIC-based principal component selection, NLL-based probabilistic scoring, and multi-layer feature representation.

3.1 DSF Agent

Although PoEM does not require annotations, manually assembling reference environment images that closely match the background of each query image is labor-intensive, limiting its scalability to new scenes. EASE (Du et al., 2025a) combines multiple foundation models to synthesize a gallery of environment images shared across all queries during inference. However, these shared references may provide only coarse semantic alignment with individual queries and may contain target-related distractors. PoEM instead aims to generate query-conditioned environment references. To improve their visual alignment with the query background while reducing target-related interference, we introduce the Describe-Sample-Filter (DSF) agent, which comprises three stages: description, sampling, and filtering. Rather than generating the final reference images directly from a single description, DSF first produces background-focused textual descriptions of the query image, then synthesizes candidate environment images from these descriptions, and finally applies visual semantic filtering to exclude candidates containing target-related content.

In the description stage, given a query image $\mathbf{q} \in \mathbb{R}^{H\times W\times 3}$ and a prompt set $\mathcal{T} \in \{t_k\}_{k=1}^{K}$, we use a multimodal large language model Φ to obtain background descriptions of $\mathbf{q}$: $p_k = \Phi(\mathbf{q}, t_k)$. In the sampling stage, we use a generative model $\mathcal{G}$ to produce an environment image for each generative prompt p_k: $\tilde{\mathbf{r}}_k = \mathcal{G}(p_k)$.

Because the generated images may still contain non-environmental components related to the camouflaged target, in the filtering stage we invoke Φ again and define a binary filtering function $\mathcal{F}(\tilde{\mathbf{r}}_k) \in \{0, 1\}$ to discriminate the generated images. Here, $\mathcal{F}(\tilde{\mathbf{r}}_k) = 0$ indicates that the image contains only environmental content. Only the generative prompts p_k corresponding to candidate samples that satisfy $\mathcal{F}(\tilde{\mathbf{r}}_k) = 0$ are considered valid and retained for the next stage. This process yields the valid prompt set

$$\mathcal{P} = \{p_k \mid \mathcal{F}(\mathcal{G}(p_k)) = 0\}. \tag{1}$$

After obtaining $\mathcal{P}$, DSF further performs multiple rounds of random sampling for each valid prompt to generate additional environment images.

$$\mathbf{r_k} = \mathcal{G}(p_k), \quad p_k \in \mathcal{P}. \tag{2}$$

The generated $\mathbf{r_k}$ still undergo visual discrimination, and only the samples satisfying $\mathcal{F}(\mathbf{r_k}) = 0$ are retained. This process is repeated until a predefined number of environment images is obtained.

Compared with the single-round generation strategy, the closed-loop mechanism of DSF reduces the probability that generated images preserve visual cues similar to those of the camouflaged target, thereby producing a cleaner and more controllable set of environment samples.

3.2 Probabilistic Modeling

Existing vision-based reference-guided methods largely depend on cross-image feature matching, where the reference image serves as an explicit template for alignment and discrepancy extraction. However, this paradigm is built on a strong assumption of cross-image correspondence and is therefore sensitive to changes in viewpoint, illumination, and background geometry. To address this limitation, we reformulate reference guidance as a problem of probabilistic environment modeling and deviation estimation. Specifically, we use reference samples to model the prior of the scene environment and identify camouflaged objects by estimating the extent to which a candidate region deviates from this prior. Under this formulation, the basis of detection shifts from cross-image consistency to environmental consistency, which improves robustness in complex scenes.

3.2.1 PPCA Formulation

To characterize the low-dimensional latent structure of reference environment samples in a high-dimensional feature space, we model the distribution of environment features with probabilistic principal component analysis (PPCA) (Tipping & Bishop, 1999). In the representation space of DINOv3 (Siméoni et al., 2025), let the environment feature be $\mathbf{x} \in \mathbb{R}^d$, where d denotes the feature dimension. PPCA assumes that $\mathbf{x}$ is generated by a linear projection of a low-dimensional latent variable with additive Gaussian noise:

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{W}\mathbf{z} + \epsilon, \tag{3}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector of the environment features, $\mathbf{W} \in \mathbb{R}^{d\times s}$ is the weight matrix, s is the dimension of the latent subspace, $\mathbf{z} \in \mathbb{R}^s$ is the low-dimensional latent variable, and $\epsilon \in \mathbb{R}^d$ is the noise term. We further assume that

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}), \quad \epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{4}$$

where $\mathbf{I}$ is the identity matrix and σ^2 is the isotropic noise variance. Under these assumptions, the feature distribution satisfies (Tipping & Bishop, 1999)

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}, \tag{5}$$

where $\mathbf{C} \in \mathbb{R}^{d\times d}$ is the covariance matrix of the environment features. In this way, PPCA represents environment features with a probabilistic model composed of a low-rank latent structure and isotropic noise.

To obtain this probabilistic model, we compute the sample covariance matrix of the environment features and perform eigendecomposition. Let the eigenvalues be $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, and let the corresponding eigenvector matrix be $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d]$. For any candidate dimension s, let $\boldsymbol{\Lambda}_s = \mathrm{diag}(\lambda_1, \ldots, \lambda_s)$ denote the diagonal matrix formed by the top s eigenvalues, and let $\mathbf{U}_s = [\mathbf{u}_1, \ldots, \mathbf{u}_s]$ denote the corresponding eigenvector matrix. For a given dimension s, the noise variance is estimated as (Tipping & Bishop, 1999)

$$\hat{\sigma}^2(s) = \frac{1}{d-s}\sum_{j=s+1}^{d}\lambda_j, \tag{6}$$

and the weight matrix is given by

$$\hat{\mathbf{W}}(s) = \mathbf{U}_s\big(\boldsymbol{\Lambda}_s - \hat{\sigma}^2(s)\mathbf{I}\big)^{1/2}. \tag{7}$$

3.2.2 Principal Components Selection

The latent dimension s determines the trade-off between model complexity and fitting capacity. In this paper, we use the Bayesian information criterion (BIC) to automatically select the number of principal components. For each candidate dimension s, the BIC is defined as

$$\mathrm{BIC}(s) = -2\log L_s + \kappa(s)\log N. \tag{8}$$

The first term, $-2\log L_s$, measures how well the model fits the data, where L_s denotes the likelihood of the PPCA model with dimension s on the environment features (see Appendix A.2). The second term, $\kappa(s)\log N$, penalizes model complexity, where N denotes the number of environment features and $\kappa(s)$ denotes the number of estimable model parameters. For PPCA, this quantity is computed as

$$\kappa(s) = ds + d + 1 - \frac{s(s-1)}{2}, \tag{9}$$

where ds, d, and 1 correspond to the numbers of parameters in $\mathbf{W}$, $\boldsymbol{\mu}$, and σ^2, respectively. The term $\frac{s(s-1)}{2}$ removes the redundant degrees of freedom induced by rotational invariance of the latent subspace. The selected number of principal components is then determined by

$$s^\star = \arg\min_{s\in\mathcal{S}} \mathrm{BIC}(s), \tag{10}$$

where $\mathcal{S} = \{1, \ldots, \min(d-1, N-2)\}$ denotes the candidate set of dimensions. The final PPCA model of the environment distribution is constructed with the corresponding $\hat{\mathbf{W}}(s^\star)$ and $\hat{\sigma}^2(s^\star)$.

3.2.3 Probabilistic Scoring

We use the negative log-likelihood (NLL) of a feature vector under the PPCA environment model to quantify its deviation from the modeled distribution. Features assigned high probability density are consistent with the reference environment, whereas those assigned low density may correspond to camouflaged objects. Under the PPCA model, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, with probability density

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{C}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]. \tag{11}$$

The corresponding negative log-likelihood is

$$-\log p(\mathbf{x}) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{C}| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}). \tag{12}$$

Omitting the constant term, we define the degree of deviation from the environment distribution as

$$f(\mathbf{x}) = \frac{1}{2}\log|\mathbf{C}| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu}). \tag{13}$$

This score plays a role similar to that of the Mahalanobis distance (see Appendix A.3). A higher value of $f(\mathbf{x})$ corresponds to a lower probability density under the modeled environment distribution, indicating that $\mathbf{x}$ is less consistent with the reference environment. We use this score to guide the localization of candidate camouflaged regions.

For camouflaged object segmentation, let $\mathbf{x}_p$ denote the patch feature at the p-th spatial location in the image. The corresponding local deviation score is defined as $\mathbf{D}_p = f(\mathbf{x}_p)$. The collection of $\mathbf{D}_p$ over all locations forms a deviation map $\mathbf{D} \in \mathbb{R}^{h\times w}$ for the image, which is used for localization and visualization of camouflaged regions. To obtain finer segmentation results, following UpGen (Du et al., 2025c), we extract prompts from $\mathbf{D}$ for SAM-based segmentation (see Appendix A.4).

3.2.4 Multi-layer Feature Representation

Existing unsupervised segmentation methods based on DINO features commonly rely on last-layer features, because these features provide strong semantic aggregation and foreground-background discrimination. Such representations tend to induce a well-structured patch similarity space, which is suitable for post-processing procedures such as clustering, graph partitioning, and similarity propagation. Although this design is simple and effective, it restricts the feature space to a single late-stage representation and may discard complementary information from adjacent layers. To address this issue, we introduce a simple and effective mechanism for multi-layer feature use, which jointly exploits multiple late-layer features without additional training or network modification.

Specifically, let $\mathbf{f}_p^l \in \mathbb{R}^{d/2}$ denote the patch feature at spatial location p from the l-th layer, where $d/2$ is the feature dimension. We select the last two layers and define the final feature representation as

$$\mathbf{x}_p = \mathrm{Concat}\left(\mathbf{f}_p^{l_1}, \mathbf{f}_p^{l_2}\right) \in \mathbb{R}^d, \tag{14}$$

where $\mathrm{Concat}(\cdot)$ denotes concatenation along the feature dimension.

Table 1: Comparison with state-of-the-art methods on four benchmark datasets. The best and second-best results are marked in **red** and **blue**, respectively. "↑/↓" indicates that higher/lower values are better. "-" denotes not available.

Method	Venue	CHAMELEON				CAMO				COD10K				NC4K			
		$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
Fully Supervised																	
SINetV2	TPAMI22	0.890	0.936	0.819	0.032	0.814	0.868	0.734	0.073	0.814	0.883	0.676	0.037	0.847	0.901	0.769	0.047
FEDER	CVPR23	0.887	0.946	0.834	0.030	0.802	0.867	0.738	0.071	0.822	0.900	0.716	0.032	0.847	0.907	0.789	0.044
CamoFormer	TPAMI24	0.910	0.957	0.865	0.022	0.872	0.929	0.831	0.046	0.869	0.932	0.786	0.023	0.892	0.939	0.847	0.030
ZoomNeXt	TPAMI24	**0.925**	0.961	**0.897**	**0.016**	0.888	0.937	0.859	0.040	**0.898**	0.945	**0.838**	**0.017**	0.900	0.940	0.865	0.028
CamoDiffusion	TPAMI25	0.907	**0.967**	0.876	0.020	0.878	0.940	0.853	0.042	0.881	0.944	0.814	0.020	0.893	0.942	0.859	0.029
ICL-Camo	TIP26	0.915	0.958	0.869	0.019	**0.893**	**0.942**	**0.860**	**0.036**	0.897	**0.950**	0.833	**0.017**	**0.912**	**0.953**	**0.879**	**0.024**
Weakly Supervised																	
WS-COD	AAAI23	0.818	**0.897**	**0.744**	**0.046**	0.735	**0.815**	0.641	**0.092**	0.733	0.832	0.576	0.049	0.775	0.855	0.688	0.063
WS-SAM	NeurIPS23	**0.820**	0.887	0.723	0.048	**0.759**	0.814	**0.667**	**0.092**	**0.803**	**0.877**	**0.680**	**0.038**	**0.829**	**0.886**	**0.757**	**0.052**
Unsupervised																	
EASE	CVPR25	0.818	0.898	0.744	0.044	0.750	0.825	0.683	0.096	0.777	0.871	0.666	0.039	0.801	0.883	0.737	0.056
UCOD-DPL	CVPR25	**0.864**	**0.931**	**0.825**	**0.031**	**0.793**	**0.862**	**0.747**	**0.077**	**0.834**	**0.916**	**0.763**	**0.031**	**0.850**	**0.923**	**0.818**	**0.043**
RISE	ICCV25	0.822	0.884	0.720	0.050	0.734	0.787	0.610	0.109	0.763	0.840	0.600	0.049	0.805	0.868	0.705	0.061
EReCu	CVPR26	0.732	0.852	0.619	0.072	0.703	0.800	0.608	0.107	0.722	0.819	0.563	0.061	0.758	0.850	0.664	0.074
DualUCOD	ICML26	0.827	0.918	0.754	0.041	0.771	0.852	0.702	0.083	0.761	0.856	0.635	0.042	0.808	0.896	0.745	0.052
Training-Free																	
GenSAM	AAAI24	0.767	0.827	0.673	0.075	0.738	0.803	0.674	0.106	0.773	0.832	0.667	0.065	0.810	0.866	0.751	0.065
MMCPF	MM24	-	-	-	-	0.749	0.820	0.680	0.101	0.733	0.803	0.592	0.066	0.767	0.827	0.681	0.083
ProMaC	NeurIPS24	0.810	0.890	0.756	0.047	0.748	0.834	0.695	0.099	0.805	0.876	0.713	0.042	0.815	0.884	0.763	0.058
UpGen	TIP25	0.821	0.874	0.757	0.048	0.793	0.842	0.737	0.089	0.802	0.852	0.703	0.053	0.836	0.883	0.785	0.056
RDVP-MSD	MM25	0.832	0.904	0.777	0.040	0.796	0.848	0.750	0.081	0.825	0.877	0.743	0.038	0.823	0.873	0.764	0.056
EASE+SAM	CVPR25	0.853	0.912	0.812	0.044	0.777	0.841	0.748	0.099	0.856	0.914	0.801	0.028	0.849	0.902	0.815	0.051
RA-COD	TIP26	0.868	0.914	0.826	**0.032**	0.823	0.884	0.781	0.063	0.863	0.926	0.798	0.024	0.872	0.924	0.833	0.034
DINOv3+SAM2	-	0.765	0.839	0.707	0.105	0.691	0.775	0.610	0.155	0.694	0.744	0.568	0.137	0.714	0.777	0.633	0.145
Qwen3+SAM2	-	0.862	0.894	0.819	0.039	0.813	0.843	0.752	0.063	0.884	0.926	0.838	**0.022**	**0.888**	0.928	**0.862**	**0.031**
Qwen3+SAM3+Category	-	0.796	0.846	0.773	0.124	0.583	0.626	0.524	0.284	0.731	0.773	0.654	0.146	0.763	0.806	0.717	0.134
Qwen3+SAM3+Bbox	-	**0.870**	0.923	**0.840**	0.040	0.816	0.866	0.770	0.066	0.882	0.929	0.837	0.025	0.884	0.925	0.859	0.036
PoEM (SAM)	-	0.866	**0.930**	0.831	0.041	**0.848**	**0.921**	**0.830**	**0.058**	**0.887**	**0.946**	**0.853**	**0.022**	0.879	**0.936**	0.857	0.036
PoEM (SAM2)	-	**0.901**	**0.959**	**0.884**	**0.023**	**0.879**	**0.938**	**0.869**	**0.042**	**0.896**	**0.953**	**0.866**	**0.019**	**0.895**	**0.946**	**0.880**	**0.029**

4 EXPERIMENT

4.1 EXPERIMENTAL SETTINGS

Datasets and Evaluation Metrics. We evaluate PoEM on four commonly used benchmarks, including CHAMELEON (76 images) (Skurowski et al., 2018), CAMO (250 images) (Le et al., 2019), COD10K (2,026 images) (Fan et al., 2020), and NC4K (4,121 images) (Lv et al., 2021). Following previous practice, structure measure (S_α) (Fan et al., 2017), mean E-measure (E_ϕ) (Fan et al., 2018), weighted F-measure (F_β^ω) (Margolin et al., 2014), and mean absolute error (M) (Perazzi et al., 2012) are adopted as evaluation metrics.

Implementation Details. For DSF agent, we adopt Qwen3-VL-4B-Instruct (Bai et al., 2025) and FLUX.2-klein-4B (Labs, 2025) as the LMM Φ and text-to-image generator $\mathcal{G}$, respectively. The prompt set $\mathcal{T}$ contains 4 different prompts used to guide the LMM in describing the environment. DSF generates 4 reference environment images for each query image. For PPCA modeling, we adopt DINOv3-ViT-L/16 (Siméoni et al., 2025) as the feature extractor and SAM2.1-hiera-large (Ravi et al., 2025) as the segmentor. More implementation details are provided in Appendix A.5.

4.2 COMPARISONS TO SOTA METHODS

Benchmark. PoEM is evaluated against state-of-the-art methods under the SAM-based segmentation setting. The comparison includes eleven training-free SAM-based methods, namely GenSAM (Hu et al., 2024a), MMCPF (Tang et al., 2024), ProMaC (Hu et al., 2024b), UpGen (Du et al., 2025c), RDVP-MSD (Yin et al., 2025), EASE+SAM (Du et al., 2025a; Kirillov et al., 2023), RA-COD (Du et al., 2026), DINOv3+SAM2 (Siméoni et al., 2025; Ravi et al., 2025), Qwen3+SAM2 (Bai et al., 2025; Ravi et al., 2025), Qwen3+SAM3+Category (Bai et al., 2025; Carion et al., 2026), and Qwen3+SAM3+Bbox (Bai et al., 2025; Carion et al., 2026). Representative supervised methods are also included for reference. More details are provided in Appendix A.6.

Comparison. Table 1 presents the quantitative comparison results. PoEM consistently outperforms existing training-free approaches across all datasets. In particular, it surpasses both text-based methods, including GenSAM (Hu et al., 2024a), ProMaC (Hu et al., 2024b), UpGen (Du et al., 2025c), and RDVP-MSD (Yin et al., 2025), as well as the vision-based method RA-COD (Du et al., 2026). These results demonstrate the effectiveness of the proposed probabilistic environment modeling framework. Furthermore, PoEM outperforms methods built on DINOv3 or Qwen3, suggesting that

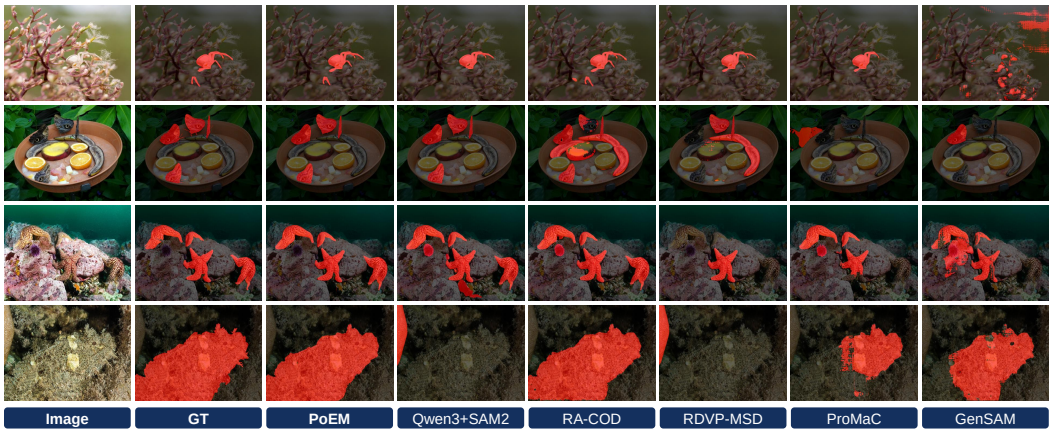


Figure 3: Qualitative comparisons with SOTA training-free methods.

Table 2: Ablation experiments on the effectiveness of DSF.

Combination	Method	COD10K				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
Qwen+FLUX	baseline	0.875	0.932	0.829	0.025	0.859	0.914	0.823	0.042
	DSF	**0.896**	**0.953**	**0.866**	**0.019**	**0.895**	**0.946**	**0.880**	**0.029**
LLaVA+FLUX	baseline	0.877	0.935	0.835	0.027	0.844	0.899	0.800	0.051
	DSF	**0.886**	**0.942**	**0.848**	**0.024**	**0.876**	**0.927**	**0.850**	**0.038**
Qwen+StableDiffusion	baseline	0.862	0.919	0.806	0.031	0.840	0.896	0.790	0.051
	DSF	**0.896**	**0.950**	**0.861**	**0.019**	**0.895**	**0.944**	**0.877**	**0.030**
LLaVA+StableDiffusion	baseline	0.870	0.926	0.819	0.030	0.840	0.898	0.794	0.053
	DSF	**0.879**	**0.933**	**0.833**	**0.026**	**0.871**	**0.924**	**0.840**	**0.040**

Table 3: Ablation experiments on the effectiveness of multi-layer feature representations.

Backbone	Layers	COD10K				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
DINOv2	22	0.883	0.939	0.837	**0.022**	0.875	0.928	0.842	0.036
	23	0.874	0.928	0.818	0.026	0.864	0.914	0.825	0.046
	Ours	**0.886**	**0.942**	**0.846**	**0.022**	**0.887**	**0.936**	**0.864**	**0.033**
DINOv3	22	0.893	0.949	0.858	**0.019**	0.884	0.938	0.861	0.032
	23	0.894	0.948	0.859	0.022	0.892	0.941	0.869	0.033
	Ours	**0.896**	**0.953**	**0.866**	**0.019**	**0.895**	**0.946**	**0.880**	**0.029**

its performance gains stem primarily from the proposed modeling paradigm, which synergistically exploits the complementary capabilities of multiple models.

To further illustrate the advantage of probabilistic environment modeling in localizing camouflaged objects, Fig. 3 presents qualitative comparisons between PoEM and representative methods. In typical camouflaged scenarios involving occlusion, multiple objects, small objects, and large objects, PoEM effectively models complex environments and accurately localizes camouflaged objects. For visualizations of the generated environment images, please refer to Appendix A.7.

4.3 Ablation Study

Ablation experiments are primarily conducted on two representative datasets, COD10K and NC4K.

Effectiveness of DSF. DSF constructs a description-sampling-filtering framework that generates reference images closely aligned with the scene context of the query image while excluding distracting objects. To evaluate the effectiveness of DSF, we use unfiltered generated images as reference images and adopt this setting as the baseline. The experiment compares multiple combinations of LMMs, including Qwen3-VL-4B-Instruct (Bai et al., 2025) and LLaVA-1.6-7B (Liu et al., 2024), with text-to-image generation models, including FLUX.2-klein-4B (Labs, 2025) and Stable-Diffusion-3.5-Large-Turbo (AI, 2024). Table 2 shows that DSF consistently outperforms the baseline across different combinations, demonstrating the effectiveness of the proposed DSF framework.

Effectiveness of Multi-Layer Feature Representation. Unlike prior methods that commonly use features from the last layer of DINO models, this paper introduces a simple strategy for exploiting multi-layer features. As shown in Table 3, multi-layer feature representation outperforms single-layer feature representation for both DINOv2 (Oquab et al., 2024) and DINOv3 (Siméoni et al., 2025). This result indicates that complementary information from different layers is important for distinguishing highly similar background and foreground features.

Effectiveness of Environment Modeling. Unlike previous methods that rely on feature matching, PoEM explores RGCOD from the perspective of probabilistic environment modeling. To assess the benefits of environment modeling over direct feature matching, we construct a feature-matching baseline that computes, for each query feature, its average cosine similarity to the top-k reference features ($k = 256$, selected empirically) and then inverts the resulting score to produce a deviation

Table 4: Ablation experiments on the effectiveness of probabilistic environment modeling.

Method	COD10K				NC4K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
feature matching	0.868	0.923	0.813	0.027	0.865	0.916	0.832	0.043
environment modeling	**0.896**	**0.953**	**0.866**	**0.019**	**0.895**	**0.946**	**0.880**	**0.029**

Table 5: Ablation experiments on the effectiveness of BIC-based principal component selection.

Method	$\kappa(s)$	COD10K				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
fixed dim ($s = 20$)	0.04M	0.868	0.915	0.800	0.028	0.866	0.915	0.823	0.043
fixed dim ($s = 2000$)	2.10M	0.896	0.952	0.865	0.019	0.895	0.945	0.879	0.029
BIC	1.50M	0.896	0.953	0.866	0.019	0.895	0.946	0.880	0.029

Table 6: Hyperparameter analysis of the number of reference environment images.

Number	COD10K				NC4K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
1	0.887	0.939	0.845	0.024	0.890	0.939	0.865	0.032
2	0.893	0.948	0.859	0.021	0.894	0.944	0.875	0.029
4	0.896	0.953	0.866	**0.019**	0.895	0.946	0.880	0.029
8	**0.897**	**0.954**	**0.868**	**0.019**	**0.898**	**0.948**	**0.883**	**0.028**

Table 7: Complexity comparison of PoEM with representative open-source methods.

Metric / Method	GenSAM	ProMaC	RDVP-MSD	PoEM
Inference time (s)	23.65	98.80	**12.43**	16.95
Parameter (B)	14.14	15.58	14.42	**12.95**
S_α on COD10K	0.773	0.805	0.825	**0.896**

map. This deviation map is then used to prompt SAM to produce refined segmentation. The results are reported in Table 4. Unlike the feature-matching baseline, PoEM does not rely on discrete prototypes. Instead, it models the distribution of the reference environment using a compact probabilistic subspace, providing a clearer statistical interpretation of the deviation scores and yielding substantial performance gains. We also evaluate alternatives to PPCA within the same environment modeling framework. Further details are provided in Appendix A.8.

Effectiveness of Principal Component Selection. To balance model performance and complexity, PoEM adopts the Bayesian information criterion (BIC) to automatically select the number of principal components. As shown in Table 5, using a small number of principal components ($s = 20$) reduces the number of parameters but limits performance, whereas using a large number of principal components ($s = 2000$) improves fitting capacity at the cost of increased model complexity. BIC-based selection achieves segmentation performance comparable to the fixed-rank setting ($s = 2000$), while reducing the reported number of estimable PPCA parameters from 2.10M to 1.50M. These results support its practical use for balancing segmentation performance and model complexity.

4.4 Further Analysis

Number of Reference Images. We examine how the number of reference environment images affects environment modeling. As shown in Table 6, PoEM maintains strong performance even when only a single reference image is used. We use four reference images by default to achieve a favorable balance between performance and complexity.

Complexity Analysis. Table 7 compares PoEM with representative open-source methods in terms of average inference time, parameter count, and performance. The inference and memory costs of PoEM mainly arise from the DSF agent, which accounts for 89% of the inference time and 96% of the parameters. The inference time of PoEM is comparable to that of RDVP-MSD (Yin et al., 2025), the fastest method, while PoEM uses fewer parameters and achieves stronger performance.

5 Conclusion

In this paper, we present PoEM for reference-guided camouflaged object detection (RGCOD). In contrast to existing text-based and vision-based methods that depend on feature matching, PoEM formulates RGCOD as a problem of probabilistic environment modeling. Specifically, PoEM models the feature distribution of reference environment images and identifies features with low likelihood under this distribution as candidate camouflaged objects. This formulation mitigates the cross-modal discrepancy encountered by text-based methods and reduces the dependence of vision-based methods on annotations. PoEM is built on two key components: a DSF agent and a probabilistic modeling framework. The DSF agent generates high-quality reference environments that are closely aligned with the environment of the query image. The probabilistic modeling framework integrates PPCA-based distribution modeling, BIC-based principal component selection, and multi-layer feature representations to produce deviation scores. Experiments demonstrate that PoEM performs environment modeling and camouflage-aware perception effectively and efficiently.

AI USE STATEMENT

In this work, we used generative AI tools for grammar checking and language polishing. We have not used generative AI tools for generating research ideas, formulating hypotheses, designing the methodology, conducting data analysis, creating figures or tables, interpreting results, or drawing conclusions, and other disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. Specifically, all AI-suggested edits were checked by the authors to ensure that the original meaning, technical accuracy, and scholarly claims were preserved. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made extensive efforts to ensure the reproducibility of our work. The architectural details of our framework, including the DSF agent and the probabilistic modeling components, are described in Section 3. Implementation details and evaluation protocols are provided in Section 4 and Appendix A.5. All datasets used in this study, namely CHAMELEON, CAMO, COD10K, and NC4K, are publicly available. Likewise, all foundation model checkpoints used in this study, including Qwen3-VL-4B-Instruct, FLUX.2-klein-4B, DINOv3-ViT-L/16, and SAM2.1-hiera-large, are publicly available.

REFERENCES

Stability AI. SD3.5. https://github.com/Stability-AI/sd3.5, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, pp. 1–42, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, pp. 1–78, 2026.

Chunyuan Chen, Weiyun Liang, Ji Du, Jing Xu, Ping Li, and Guiling Wang. Learn from examples: In-context learning for camouflaged object detection. *IEEE Transactions on Image Processing*, 35:3793–3806, 2026a.

Huafeng Chen, Chenguang Zhu, Yueming Lyu, and Caifeng Shan. Beyond weak supervision: Mllms-guided graded knowledge distillation for unsupervised camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 27547–27557, 2026b.

Ji Du, Fangwei Hao, Mingyang Yu, Desheng Kong, Jiesheng Wu, Bin Wang, Jing Xu, and Ping Li. Shift the lens: Environment-aware unsupervised camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19271–19282, 2025a.

Ji Du, Xin Wang, Fangwei Hao, Mingyang Yu, Chunyuan Chen, Jiesheng Wu, Bin Wang, Jing Xu, and Ping Li. Beyond single images: Retrieval self-augmented unsupervised camouflaged object detection. In *IEEE International Conference on Computer Vision*, pp. 22131–22142, 2025b.

Ji Du, Jiesheng Wu, Desheng Kong, Weiyun Liang, Fangwei Hao, Jing Xu, Bin Wang, Guiling Wang, and Ping Li. UpGen: Unleashing potential of foundation models for training-free camouflage detection via generative models. *IEEE Transactions on Image Processing*, 34:5400–5413, 2025c.

Ji Du, Jiesheng Wu, Desheng Kong, Fangwei Hao, Jing Xu, and Ping Li. Ra-cod: Retrieval-augmented camouflaged object detection. *IEEE Transactions on Image Processing*, 35:5253–5265, 2026.

Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-Measure: A new way to evaluate foreground maps. In *IEEE International Conference on Computer Vision*, pp. 4558–4567, 2017.

Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *International Joint Conference on Artificial Intelligence*, pp. 698–704, 2018.

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2774–2784, 2020.

Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6024–6042, 2022.

Chao Hao, Zitong Yu, Xin Liu, Jun Xu, Huanjing Yue, and Jingyu Yang. A simple yet effective network based on vision transformer for camouflaged object and salient object detection. *IEEE Transactions on Image Processing*, 34:608–622, 2025.

Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Camouflaged object detection with feature decomposition and edge reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22046–22055, 2023a.

Chunming He, Kai Li, Yachao Zhang, Guoxia Xu, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Weakly-supervised concealed object segmentation with SAM-based pseudo labeling and multi-scale feature grouping. In *Advances in Neural Information Processing Systems*, pp. 30726–30737, 2023b.

Chunming He, Kai Li, Yachao Zhang, Yulun Zhang, Zhenhua Guo, Xiu Li, Martin Danelljan, and Fisher Yu. Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects. In *International Conference on Learning Representations*, pp. 1–10, 2024.

Chunming He, Kai Li, Yachao Zhang, Ziyun Yang, Youwei Pang, Longxiang Tang, Chengyu Fang, Yulun Zhang, Linghe Kong, Xiu Li, and Sina Farsiu. Segment concealed objects with incomplete supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025a.

Chunming He, Rihan Zhang, Fengyang Xiao, Chenyu Fang, Longxiang Tang, Yulun Zhang, Linghe Kong, Deng-Ping Fan, Kai Li, and Sina Farsiu. RUN: Reversible unfolding network for concealed object segmentation. In *International Conference on Machine Learning*, pp. 1–14, 2025b.

Chunming He, Rihan Zhang, Fengyang Xiao, Dingming Zhang, Zhiwen Cao, and Sina Farsiu. Refining context-entangled content segmentation via curriculum selection and anti-curriculum promotion. In *International Conference on Machine Learning*, pp. 1–13, 2026.

Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson W.H. Lau. Weakly-supervised camouflaged object detection with scribble annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 781–789, 2023c.

Jian Hu, Jiayi Lin, Shaogang Gong, and Weitong Cai. 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, 2024a.

Jian Hu, Jiayi Lin, Junchi Yan, and Shaogang Gong. Leveraging hallucinations to reduce manual prompt dependency in promptable segmentation. In *Advances in Neural Information Processing Systems*, pp. 1–13, 2024b.

Shuo Jiang, Gaojia Zhang, Min Tan, Yufei Yin, and Gang Pan. Erecu: Pseudo-label evolution fusion and refinement with multi-cue learning for unsupervised camouflage detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25547–25556, 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE International Conference on Computer Vision*, pp. 3992–4003, 2023.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, 184:45–56, 2019.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Pingzhu Liu, Chunming He, Zunnan Xu, Chao Hao, Bo Zhao, Xingyu Shao, Jun Zhou, Zitong Yu, and Xiu Li. Unsupervised camouflaged object detection with dual-eigenvector spectral pseudo-labeling and contrastive refinement. In *International Conference on Machine Learning*, 2026.

Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11586–11596, 2021.

Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2014.

Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen, Vinh-Tiep Nguyen, Thanh Duc Ngo, Thanh-Toan Do, Minh-Triet Tran, and Tam V. Nguyen. The art of camouflage: Few-shot learning for animal detection and segmentation. *IEEE Access*, 12:103488–103503, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024:1–32, 2024.

Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2150–2160, 2022.

Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. ZoomNeXt: A unified collaborative pyramid network for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9205–9220, 2024.

Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–740, 2012.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, pp. 1–17, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, pp. 1–67, 2025.

Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018.

Ke Sun, Zhongxi Chen, Xianming Lin, Xiaoshuai Sun, Hong Liu, and Rongrong Ji. Conditional diffusion models for camouflaged and salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2833–2848, 2025.

Lv Tang, Peng-Tao Jiang, Zhi-Hao Shen, Hao Zhang, Jin-Wei Chen, and Bo Li. Chain of Visual Perception: Harnessing multimodal large language models for zero-shot camouflaged object detection. In *ACM International Conference on Multimedia*, pp. 8805–8814, 2024.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14543–14553, 2022.

Weiqi Yan, Lvhai Chen, Huaijia Kou, Shengchuan Zhang, Yan Zhang, and Liujuan Cao. UCOD-DPL: Unsupervised camouflaged object detection via dynamic pseudo-label learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 30365–30375, 2025.

Yilong Yang, Jianxin Tian, Shengchuan Zhang, and Liujuan Cao. Discover, segment, and select: A progressive mechanism for zero-shot camouflaged object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34745–34754, 2026.

Bowen Yin, Xuying Zhang, Deng-Ping Fan, Shaohui Jiao, Ming-Ming Cheng, Luc Van Gool, and Qibin Hou. CamoFormer: Masked separable attention for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10362–10374, 2024.

Chao Yin, Hao Li, Kequan Yang, Jide Li, Pinpin Zhu, and Xiaoqiang Li. Stepwise decomposition and dual-stream focus: A novel approach for training-free camouflaged object segmentation. In *ACM International Conference on Multimedia*, pp. 3741–3750, 2025.

Zihou Zhang, Hao Li, Zhengwei Yang, Zechao Hu, Liang Li, and Zheng Wang. From language to instance: Generative visual prompting for zero-shot camouflaged object detection. In *ACM International Conference on Multimedia*, pp. 382–391, 2025.

A APPENDIX

A.1 PPCA Covariance Regularization and Invertibility 14
A.2 Working-Independence Approximation for BIC-Based Rank Selection 14
A.3 NLL-based Score and Mahalanobis Distance 14
A.4 Prompt Extraction 15
A.5 More Implementation Details 15
A.6 Baseline Construction and Supervised Methods 15
A.7 More Visualizations 16
A.8 Different Environment Modeling Methods 16
A.9 Failure Case Analysis 18

A.1 PPCA COVARIANCE REGULARIZATION AND INVERTIBILITY

For numerical stability, we impose a lower bound on the noise variance estimated in Eq. (6):

$$\tilde{\sigma}^2 = \max(\hat{\sigma}^2, \varepsilon), \tag{15}$$

where $\varepsilon = 10^{-6}$. The resulting regularized covariance is

$$\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \tilde{\sigma}^2\mathbf{I}. \tag{16}$$

Since $\mathbf{W}\mathbf{W}^\top$ is positive semidefinite, we have

$$\lambda_{\min}(\mathbf{C}) \geq \tilde{\sigma}^2 \geq \varepsilon > 0. \tag{17}$$

Thus, $\mathbf{C}$ is positive definite and invertible, even when the sample covariance is rank-deficient.

A.2 WORKING-INDEPENDENCE APPROXIMATION FOR BIC-BASED RANK SELECTION

The BIC-based criterion in Eq. (8) is evaluated using the pooled reference patch features. For each candidate latent dimension s, we compute

$$\log L_s = \sum_{i=1}^{N} \log \mathcal{N}\left(\mathbf{x}_i; \hat{\boldsymbol{\mu}}, \hat{\mathbf{C}}_s\right), \tag{18}$$

where N is the total number of reference patch vectors, $\hat{\boldsymbol{\mu}}$ is their sample mean, and $\hat{\mathbf{C}}_s$ is the fitted PPCA covariance.

This calculation adopts a working-independence approximation across patches. In practice, patch features from the same reference image may exhibit spatial dependence. Although the PPCA covariance captures correlations among the dimensions of each feature vector, it does not explicitly model dependencies between different patches. Consequently, N denotes the number of pooled patch vectors rather than an effective number of independent observations.

We therefore interpret Eq. (8) as a practical BIC-based criterion for selecting the covariance rank, without claiming that classical BIC model-selection guarantees hold for these dependent samples. Its practical utility is evaluated through the segmentation performance and model complexity comparisons in Table 5. Explicitly accounting for spatial dependence in rank selection remains a direction for future work.

A.3 NLL-BASED SCORE AND MAHALANOBIS DISTANCE

For each query image, all patch features are evaluated under the same environment model, with a shared mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$. Consequently, the log-determinant term in Eq. (13) is constant across spatial locations. Let $\mathbf{D}_p = f(\mathbf{x}_p)$ denote the deviation score at location p, and let $m_p =$

$(\mathbf{x}_p - \boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x}_p - \boldsymbol{\mu})$ denote the corresponding squared Mahalanobis distance. For a nonconstant score map, min–max normalization gives

$$\bar{\mathbf{D}}_p = \frac{\mathbf{D}_p - \min_q \mathbf{D}_q}{\max_q \mathbf{D}_q - \min_q \mathbf{D}_q} = \frac{m_p - \min_q m_q}{\max_q m_q - \min_q m_q}, \tag{19}$$

where q indexes spatial locations in the query image.

Thus, the normalized NLL map is equivalent to a normalized squared Mahalanobis distance map under the PPCA covariance. Localization therefore relies on relative, covariance-weighted deviations within each query image. Higher scores indicate lower compatibility with the modeled environment and serve as cues for identifying candidate camouflaged regions.

A.4 Prompt Extraction

Following UpGen (Du et al., 2025c), we extract point and box prompts from min-max-normalized deviation maps to guide SAM in producing refined segmentation masks. We first binarize each deviation map using a global threshold. For each connected component in the resulting binary map, we use its minimum bounding rectangle as a box prompt. Within each box, we select the locations with the highest and lowest deviation scores as the positive and negative point prompts, respectively. To suppress noisy regions, we retain only boxes whose maximum deviation score exceeds a local threshold. The global and local thresholds are fixed at 0.4 and 0.7, respectively, across all datasets.

A.5 More Implementation Details

For DSF agent, we use Qwen3-VL-4B-Instruct for both environment description and candidate filtering, and FLUX.2-klein-4B for image generation. Each generator call produces one 1024×1024 image using four inference steps and a guidance scale of 1.0. On CUDA devices, the FLUX pipeline runs in bfloat16 precision, while Qwen uses automatic dtype selection and device placement. Each description or filtering response is limited to 512 newly generated tokens. Other language-model decoding parameters are left at the defaults specified by the loaded model's generation configuration.

The prompt set $\mathcal{T}$ contains four instruction templates for describing the query environment:

- *Describe the environment in this image without focusing on the main object. Your response will be directly used as input for a text-to-image model to generate an image.*
- *Describe the environment of this image while ignoring any animal or human. Your response will be directly used as input for a text-to-image model to generate an image.*
- *Provide a description of the surrounding environment in the image only. Your response will be directly used as input for a text-to-image model to generate an image.*
- *Please just focus on and analyze the background environment shown in the image. Your response will be directly used as input for a text-to-image model to generate an image.*

The filtering prompt is: "*Is there any animal or human in the image? Answer yes or no.*" Each filtering call receives the generated candidate and this instruction. The response is converted to lowercase, and a candidate is rejected if the substring "yes" is present; otherwise, it passes the filter.

For PPCA modeling, following previous practice (Hu et al., 2024a;b; Yin et al., 2025), both query and reference images are resized to 1024×1024. We extract features from the last two layers of DINOv3-ViT-L/16, each producing a 64×64 grid of 1024-dimensional patch features. Concatenating these features yields 4096 patch vectors of dimension $d = 2048$ per image. Each concatenated vector is independently ℓ_2-normalized for both query and reference images.

A.6 Baseline Construction and Supervised Methods

For existing training-free methods, we use either prediction maps provided by the authors or predictions generated using their official implementations. All such predictions are evaluated using the same evaluation framework and metric implementations to ensure consistency.

We next describe the newly constructed training-free baselines evaluated in Table 1. For DINOv3+SAM2, we follow the official DINOv3 implementation and use TokenCut (Wang et al., 2022), an unsupervised object discovery method, to generate initial segmentation masks. We then extract prompts from these masks and provide them to SAM2 to obtain refined segmentations.

For Qwen3+SAM2 and Qwen3+SAM3+Bbx, we prompt Qwen3-VL-4B-Instruct to predict bounding boxes for candidate camouflaged objects. We then provide these boxes as prompts to SAM2 and SAM3, respectively, to obtain the final segmentation masks. Empirically, prompts that omit camouflage-related concepts yield better performance than those that include them. A possible explanation is that Qwen3 incorrectly judges some camouflaged targets to be non-camouflaged and consequently excludes them when explicitly instructed to detect camouflaged objects. We therefore use the following prompt:

- *You are an expert detector. Detect every animal or human in this image. Only return targets that are actually animals or humans. Return JSON only, with no explanation and no markdown. The output must be a JSON array. Each item must have the format {"bbox_2d": [x1, y1, x2, y2], "label": "animal"} or {"bbox_2d": [x1, y1, x2, y2], "label": "human"}. Include all distinct targets. If there is no animal or human, return [].*

For Qwen3+SAM3+Category, we prompt Qwen3-VL-4B-Instruct to identify the category of the camouflaged object and provide the predicted category as a text prompt to SAM3 to obtain the final segmentation. When SAM3 fails to localize the target using its predicted category, we additionally query Qwen3-VL-4B-Instruct to identify the environment and the categories of surrounding non-camouflaged objects. We then use these predictions as text prompts for SAM3 to segment the environment and surrounding objects. Taking the complement of the union of the resulting masks yields an indirect estimate of the target mask.

We use the following prompt to identify the target category:

- *What is the creature hiding in the environment? Answer with a single word.*

To identify the environment, we use:

- *What is the environment of this image? Answer this question with a single word or phrase.*

To identify the category of a surrounding object, we use:

- *What is the object next to the animal or person in the image? Answer this question with a single word or phrase.*

We also include representative methods under different supervision settings for reference, including fully supervised methods (SINetV2 (Fan et al., 2022), FEDER (He et al., 2023a), CamoFormer (Yin et al., 2024), ZoomNeXt (Pang et al., 2024), CamoDiffusion (Sun et al., 2025), and ICL-Camo (Chen et al., 2026a)), weakly supervised methods (WS-COD (He et al., 2023c) and WS-SAM (He et al., 2023b)), and unsupervised methods (EASE (Du et al., 2025a), UCOD-DPL (Yan et al., 2025), RISE (Du et al., 2025b), EReCu (Jiang et al., 2026), and DualUCOD (Liu et al., 2026)).

A.7 More Visualizations

We provide visualizations of the generated reference environment images and segmentation masks for underwater, terrestrial, and human-centered scenarios in Fig. 4, Fig. 5, and Fig. 6, respectively. These results further demonstrate that DSF generates reference environment images that closely align with the environmental context of the corresponding query images while reducing target-related content.

A.8 Different Environment Modeling Methods

We evaluate several alternatives to PPCA for environment modeling, including principal component analysis (PCA) with reconstruction-based scoring, factor analysis (FA), one-class support vector machines (SVM), k-nearest neighbors (KNN), and k-means clustering (K-means). We also include two



Figure 4: More visualizations in underwater scenarios.

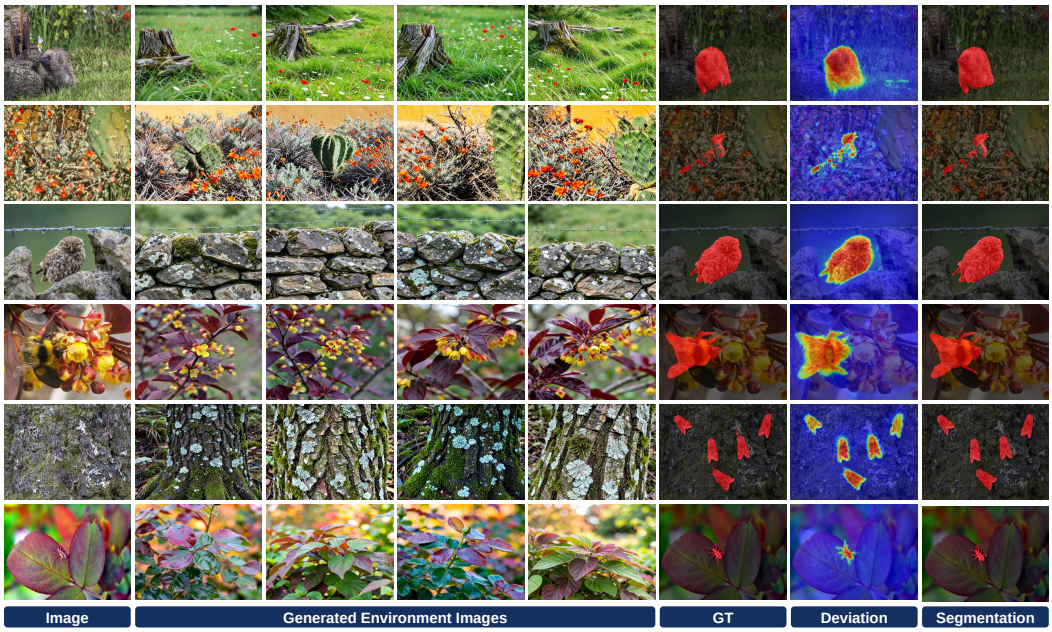


Figure 5: More visualizations in terrestrial scenarios.

Gaussian baselines with different covariance structures: full-covariance Gaussian (FG) and diagonal Gaussian (DG) models. All methods use the same reference galleries. As shown in Table 8, the evaluated methods achieve competitive performance, supporting the effectiveness of environment modeling. PPCA achieves the best performance among the evaluated methods. It models background features using a low-dimensional latent structure with isotropic residual noise, capturing dominant feature correlations while constraining covariance complexity. This formulation provides a compact background representation with a clear probabilistic interpretation.

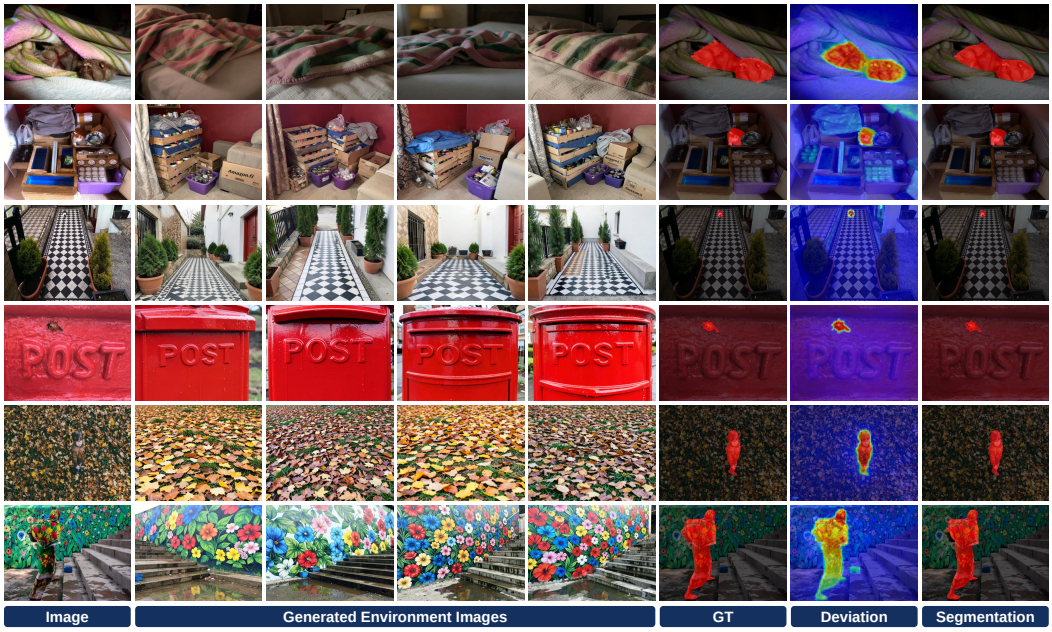


Figure 6: More visualizations in human-centered scenarios.

Table 8: Ablation experiments on different environment modeling methods.

Method	CHAMELEON				CAMO				COD10K				NC4K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
PCA	0.882	0.940	0.857	0.040	0.867	0.927	0.844	0.048	0.887	0.938	0.838	0.022	0.883	0.933	0.851	0.034
FA	0.893	0.954	0.869	0.027	0.874	0.931	0.855	0.046	0.893	0.946	0.855	0.020	**0.895**	0.942	0.874	0.030
SVM	0.865	0.923	0.839	0.057	0.829	0.888	0.801	0.075	0.866	0.912	0.800	0.034	0.848	0.897	0.799	0.057
KNN	0.889	0.951	0.859	0.029	0.844	0.900	0.817	0.063	0.865	0.911	0.799	0.034	0.863	0.911	0.820	0.048
Kmeans	0.879	0.938	0.836	0.034	0.811	0.864	0.772	0.089	0.829	0.871	0.736	0.052	0.835	0.880	0.774	0.064
FG	0.893	0.954	0.875	0.024	0.872	0.928	0.861	0.047	0.887	0.945	0.857	0.025	0.890	0.939	0.876	0.033
DG	0.886	0.945	0.857	0.029	0.829	0.888	0.788	0.069	0.860	0.906	0.793	0.033	0.865	0.912	0.826	0.045
PPCA	**0.901**	**0.959**	**0.884**	**0.023**	**0.879**	**0.938**	**0.869**	**0.042**	**0.896**	**0.953**	**0.866**	**0.019**	**0.895**	**0.946**	**0.880**	**0.029**

A.9 Failure Case Analysis

We present several failure cases of PoEM in Fig. 7. In the first two rows, PoEM successfully localizes the main target regions but does not fully recover fine structures. This limitation primarily reflects constraints in SAM's segmentation capabilities and may be compounded by the loss of spatial detail in DINOv3's patch-based representation. In the last two rows, visual similarity between the generated environment images and the targets introduces ambiguity into environment modeling. In such cases, the deviation maps may exhibit excessive or insufficient activation. These cases highlight the need to improve both the recovery of fine structures and the robustness of environment modeling.

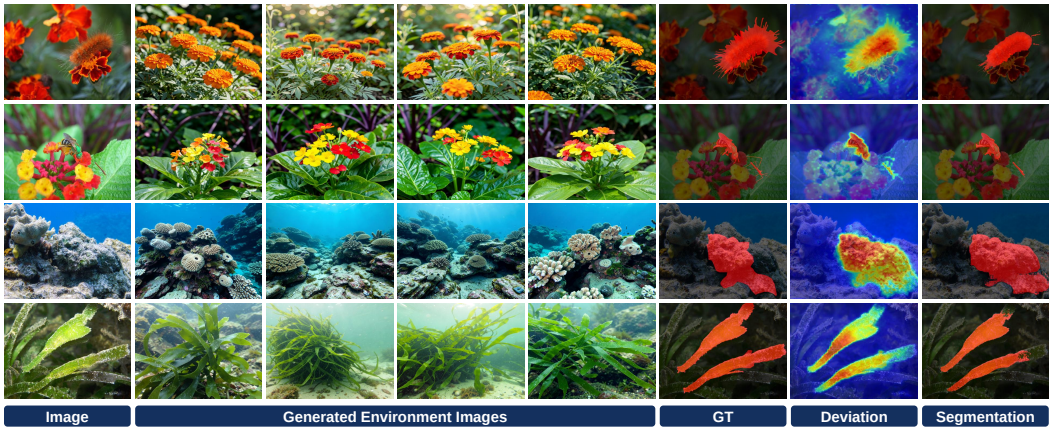


Figure 7: PoEM may fail to fully capture fine-grained structures and may produce over-activated or under-activated deviation maps in challenging cases.